



Diplomarbeit

"Entwickeln und Bewerten Fehler erkennender Programmbausteine in speicherprogrammierbaren Steuerungen (SPS) zur Erhöhung deren Sicherheit"

Fachhochschule Bonn-Rhein-Sieg

Fachbereich EMT

Studiengang Maschinenbau / Mechatronik

Grantham-Allee 20

53757 Sankt Augustin

Vorgelegt von:

Björn Ostermann

Erstprüfer:	**Prof. Dr. Josef Vollmer**
Zweitprüfer:	**Prof. Dr. Wolfgang Joppich**

Sankt Augustin, 14. Juli 2006

„Das Einsatzgebiet der Standard-SPS im Bereich sicherheitstechnischer Steuerungen ist durch die Einstufung in die unterste Sicherheitsstufe (Kategorie 1 – Performance Level „a“) stark eingeschränkt und nach der heute gültigen DIN EN 954-1 nur unter Anwendung zusätzlicher sicherheitstechnischer Maßnahmen möglich. Die Sicherheits-SPS kann zwar in allen Bereichen der sicherheitstechnischen Steuerungen eingesetzt werden (Kategorie 4 – Performance Level „e“), ist aber […] durch ihren hohen Preis in den niedrigen Sicherheitsstufen unwirtschaftlich.“

[aus Kapitel 7.3]

Danksagung

Zum Gelingen dieser Arbeit haben mehrere Personen beigetragen, bei denen ich mich an dieser Stelle bedanken will.

Dies sind die Mitarbeiter des BGIA, die mir durch Fachdiskussionen und Hinweisen auf Literatur oft weitergeholfen haben. Insbesondere bedanke ich mich bei Dr. Michael Huelke für die ausgezeichnete und intensive Betreuung sowie bei Dr. Michael Schaefer, für die Möglichkeit diese Diplomarbeit beim BGIA durchführen zu können.

Mein Dank gilt ganz besonders auch meinen betreuenden Professoren Prof. Dr. Josef Vollmer und Prof. Dr. Wolfgang Joppich.

Weiterhin gilt mein Dank meinen Korrekturlesern und hier besonders meinem Vater, Dipl.-Ing. Hans-J. Ostermann, für die zahlreichen Literaturhinweise und Fachdiskussionen im Bereich der Maschinenrichtlinie.

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Niederkassel, 10.07.2006

Inhaltsverzeichnis

1 Einleitung

1.1 Thema der Diplomarbeit

Die Diplomarbeit beschäftigt sich mit speicherprogrammierbaren Steuerungen (SPS). Diese werden in der Industrie zum automatischen Steuern von Maschinen eingesetzt. Durch fehlerhafte Programmierung wie auch durch Fehler in der Hardware der SPS können Gefahren für Mensch und Material entstehen.

Das Thema der Diplomarbeit ist die Verbesserung der Sicherheit von SPSen durch sicherheitstechnische Funktionsbausteine, die den Programmen der SPSen hinzugefügt werden und die die korrekte Funktion der Hardware der SPSen überwachen.

In diesem Zusammenhang werden die rechtlichen Grundlagen behandelt, die ein bestimmtes Maß an Sicherheit von Steuerungen fordern, sowie die technischen Regeln (Normen), anhand derer diese Sicherheit beurteilt werden kann.

1.2 Motivation

Das Bedürfnis solche Funktionsbausteine in einer Standard-SPS einzusetzen und damit deren Sicherheit zu erhöhen, entsteht aus der aktuellen Marktsituation. Zur Zeit sind am Markt zum Einen Standard-SPSen verfügbar, Steuerungsbauteile, die wenigen bis gar keinen Sicherheitsanforderungen genügen, und zum Anderen Sicherheits-SPSen. Diese bieten zwar maximale Sicherheit, sind dafür aber verhältnismäßig teuer in der Anschaffung. Dazwischen liegt ein breites Feld an unterschiedlichen Sicherheitsanforderungen an Steuerungen. Dieser Bedarf kann zwar sicherheitstechnisch mit den Sicherheits-SPSen abgedeckt werden, dies ist aus wirtschaftlichen Gesichtspunkten aber nicht optimal.

Das Ziel dieser Arbeit ist es, Funktionsbausteine für eine Standard-SPS zu entwickeln, die mittels der darin enthaltenen Testroutinen den o. a. Bereich sicherheitstechnisch optimal abdecken und dabei eine wirtschaftliche Lösung bieten. Diese Standard-SPS (in dieser Diplomarbeit als Standard-SPS mit Diagnosefunktionen bezeichnet) kann mit diesen Tests nicht das Sicherheitsniveau der Sicherheits-SPS erreichen. Dies ist aber bei den Sicherheitsanforderungen vieler Anwendungen auch nicht erforderlich.

1.3 Lösungsweg

Im Rahmen dieser Diplomarbeit wurden Testverfahren zur Gewährleistung der sicherheitstechnischen Überwachung einer SPS erarbeitet, die ein Anwender einer SPS als Funktionsbausteine in das Programm der SPS implementieren kann. Soweit nutzbare Fehlererkennungsroutinen in anderen Bereichen bekannt sind, wurden diese bei der Lösungssuche berücksichtigt. (siehe Kapitel 1.4) Die Arbeitsweise der erarbeiteten Testverfahren wird erläutert und deren Wirkungsgrad aufgezeigt.

Um die Sicherheit eines Steuerungsbauteils zu beurteilen, wird in der Industrie grundsätzlich die nach der gesetzlich vorgegebenen europäischen Maschinenrichtlinie [26] harmonisierte Norm DIN EN 954-1 [14] herangezogen. Neben der Berücksichtigung dieser Norm geht diese Diplomarbeit

zusätzlich auf die Vor-Norm prEN ISO 13849-1 [15] ein, die eine wesentlich feinere Einschätzung der erreichten Sicherheit zulässt und die in naher Zukunft auch die DIN EN 954-1 ablösen wird.

Diese feinere Einschätzung der erreichten Sicherheit ergibt sich durch die Berücksichtigung der Ausfallwahrscheinlichkeiten und der Diagnosedeckungsgraden (Diagnostic Coverage – DC – Wahrscheinlichkeit des Entdeckens eines Fehlers). Aus diesem Grund wird in dieser Arbeit zu jedem Test angegeben, welcher DC mit dem jeweiligen Test erreicht werden kann.

Zum Abschluss der Diplomarbeit wird eine Beispiel-SPS mit integrierten Testverfahren anhand der o. a. Normen bewertet und die erreichte Verbesserung aufgezeigt.

1.4 Fehlererkennung in anderen Bereichen

Das Problem, dass ein System, welches aus elektronischen Komponenten besteht, nicht auf Dauer fehlerfrei arbeiten kann, ist so alt wie die Elektronik. Aus diesem Grund existieren in anderen Bereichen bereits zahlreiche Lösungen für Selbsttests von Steuerungen.

Elektrische Schaltungen, die fehlersicher sein müssen, werden z. B. redundant aufgebaut. Bei Systemen, die Prozessoren enthalten, bietet sich dagegen der Selbsttest als wirtschaftliche Alternative zum redundanten Aufbau an. Ein einkanaliges System mit Selbsttest kann allerdings nie die gleiche Sicherheit wie eine redundante Schaltung bieten.

Für diese Diplomarbeit wurden vor allem bereits umgesetzte und auf ihre Funktionalität hin geprüfte Testverfahren für den PC und den Mikroprozessor betrachtet, um hieraus Erkenntnisse für die in dieser Diplomarbeit entwickelten Tests zu gewinnen.

Beim PC wurde der Speichertest Memtest86 betrachtet, beim Mikroprozessor der Inhalt des BGIA Reports #/2006 [1]. Außerdem wurden die Angaben aus dem allgemein gehaltenen Buch [4] über Programmlaufüberwachungen und der Norm DIN EN 61508 [17] über die funktionale Sicherheit elektronischer Systeme berücksichtigt.

2 Aufgabenstellung

2.1 Ziel

Das Ziel der Diplomarbeit ist die Erhöhung der Steuerungskategorie einer Standard-SPS (siehe Kapitel 6.1) nach DIN EN 954-1 [14] bzw. die Erhöhung ihres Performance Levels nach prEN ISO 13849-1 [15].

Zu diesem Zweck soll die Standard-SPS durch Programmierung von Fehler erkennenden Funktionsbausteinen (siehe Kapitel 8.2.2) und Verschaltung mit sich selbst und einer zweiten Standard-SPS so erweitert werden, dass zufällige Hardwarefehler, die während des Betriebs auftauchen können, erkannt werden.

Zur Einordnung in die Sicherheitsstufen der oben erwähnten Normen soll der erreichte Diagnosedeckungsgrad der Funktionsbausteine bewertet werden. Der Quelltext dieser Funktionsbausteine liegt dieser Diplomarbeit als Anlage (Anhang II) bei.

Im praktischen Teil dieser Diplomarbeit wurde keine reale SPS verwendet. Als SPS wurde die Soft-SPS (siehe Kapitel 6.1.2) von CoDeSys (Code Development System) genutzt. Die Funktionsbausteine sind im Hinblick auf eine Schrank-SPS (siehe Kapitel 6.1.1) entwickelt, die die Sprache „Anweisungsliste“ (AWL – siehe Kapitel 8.3.1) nach der Norm DIN EN 61131-3 [16] unterstützt.

2.2 Anforderungen an die Funktionsbausteine

Im Rahmen der Diplomarbeit wurden folgende Anforderungen an die zu programmierenden Funktionsbausteine in Zusammenarbeit mit dem BGIA festgelegt:

- Diese sollen in AWL nach DIN EN 61131-3 [16] programmiert werden, um so einerseits möglichst portabel und andererseits möglicht schnell und hardwarenah zu sein.
- Beim Auftritt eines Fehlers sollen sie die SPS in einen sicheren Zustand versetzen.
- Sie sollen vom Anwender möglichst einfach in das eigene Programm übertragbar sein und keine Eingriffe in die Architektur der SPS erfordern.
- Die von ihnen durchgeführten Tests sollen, wo möglich, zum Programmstart komplett und während der Laufzeit zyklisch ablaufen.
- Der Anwender soll die Möglichkeit bekommen durch das Übergeben von Parametern einzustellen, wie viel Anteil vom Gesamttest in einem Durchlauf abgearbeitet wird.
- Im Einzelnen sollen sie die Funktion des Prozessors, des Speichers und der Ausgabe testen.
- Zum Testen des Prozessors sollen sie die grundlegenden logischen und arithmetischen Funktionen testen sowie eine Überwachung des Programmablaufs realisieren.
- Zum Testen des Speichers sollen sie eine zyklische Prüfung durchführen. Wo dies nicht möglich ist, soll der Speicher auf Lesbarkeit und Schreibbarkeit getestet werden.

- Der Ausgabetest soll unter Annahme einer realisierten Rückkopplung der Ausgänge in die Eingänge geschehen.

2.3 Bewertung der erreichten Sicherheit

Anhand typischer Kennwerte einer Schrank-SPS soll die erreichte Verbesserung der Sicherheit im Hinblick auf die in Kapitel 2.1 erwähnten Normen aufgezeigt werden.

3 Aufbau der Diplomarbeit

Die Diplomarbeit ist in fünf Sinnabschnitte gegliedert.

Sinnabschnitt 1

Der erste Sinnabschnitt umfasst die Kapitel 1 bis 3 und beinhaltet die Einleitung, die Aufgabenstellung und den Aufbau der Diplomarbeit selbst.

Sinnabschnitt 2

Im zweiten Sinnabschnitt (Kapitel 4 und 5) werden die rechtlichen Grundlagen für das in Verkehr bringen einer SPS mit Diagnosefunktion im europäischen Wirtschaftsraum (EWR) behandelt. Hieraus ergibt sich die Notwendigkeit, eine solche SPS nach der Norm DIN EN 954-1 bzw. prEN ISO 13849-1 zu bewerten. Auch dieses Thema wird in diesem Sinnabschnitt behandelt.

Es wird dargelegt, welche Gesetze und Verordnungen für Steuerungen wichtig sind und welche technischen Anforderungen aus diesen und den harmonisierten Normen entstehen. Es wird weiterhin dargestellt, welchen Stellenwert Verordnungen, Gesetze und Normen haben.

Sinnabschnitt 3

Der dritte Sinnabschnitt (Kapitel 6 bis 8) enthält technische Grundlagen zur SPS und deren Sicherheitseinstufung. Es wird der interne Hardware Aufbau einer SPS sowie ihre generelle Arbeitsweise und ihre Programmierung nach DIN EN 61131-3 beschrieben. Die auf dem Markt vorhandenen SPS-Alternativen Standard-SPS und Sicherheits-SPS werden nach den oben erwähnten Normen DIN EN 954-1 und prEN ISO 13849-1 hinsichtlich ihrer Sicherheitseinstufung bewertet um den Stand der Technik bei SPSen aufzuzeigen.

Sinnabschnitt 4

Der vierte Sinnabschnitt (Kapitel 9 bis 12) enthält eine Beschreibung möglicher Hardwarefehler einer SPS und die Möglichkeiten ihrer Erkennung. Weiterhin wird die Entwicklung von Fehlererkennungsroutinen für eine SPS mit ihren jeweiligen Diagnosedeckungsgraden behandelt. Der Sinnabschnitt schließt mit der an den Vorgaben der Normen DIN EN 954-1 und prEN ISO 13849-1 gemessenen Verbesserung der Sicherheit einer Standard-SPS mit den aufgezeigten Verfahren.

Sinnabschnitt 5

Im letzten Sinnabschnitt (Kapitel 13) wird das Ergebnis der Diplomarbeit zusammengefasst und es werden Ausblicke auf weiterführende Arbeiten gegeben.

Verzeichnisse

Die Kapitel 14 bis 16 beinhalten die Verzeichnisse der Abbildungen, der Abkürzungen und der verwendeten Literaturquellen.

Anhänge – Anlage

Der Diplomarbeit sind zwei Anhänge und eine Anlage beigefügt:

- Anhang I enthält weiterführende Informationen zum Europäischen Binnenmarkt.
- Anhang II enthält den Quelltext als Klartext der im Rahmen dieser Diplomarbeit programmierten Funktionsbausteine. Dieser Anhang wird aus praktischen Gründen – wegen des großen Umfangs – auf der u. a. CD beigefügt. (PDF-Datei)
- Anhang III enthält den Quelltext als CoDeSys-Quellcode. Dieser befindet sich ebenfalls auf der u. a. CD.
- Als Anlage liegt der Diplomarbeit eine CD mit den o. a. Quelltexten bei.

4 Rechtliche Grundlagen für die Sicherheit von Steuerungen

SPSen sind Produkte des freien Warenverkehrs. Sie unterliegen damit den staatlichen Vorschriften, die das in Verkehr bringen solcher Produkte regeln. Diese staatlichen Vorschriften fußen im Europäischen Wirtschaftsraum (EWR) auf einheitlichen europäischen Regelungen in denen Anforderungen an die Sicherheit und den Gesundheitsschutz formuliert sind.

Der Gesamtzusammenhang der europäischen Regelungen und deren Auswirkung auf das in Verkehr bringen einer SPS mit Diagnosefunktionen wird in diesem Kapitel dargestellt. Hierbei ist besonders die europäische Maschinenrichtlinie [26] (MRL) zu beachten, die in ihrem technischen Anhang konkrete Anforderungen an die sicherheitstechnische Funktion von Steuerungen und damit auch an eine SPS mit Diagnosefunktionen stellt.

4.1 Europäischer Binnenmarkt

Eines der großen Ziele der Europäischen Gemeinschaft (EG) ist der freie Warenverkehr zwischen den Staaten dieser Gemeinschaft. Dieses Ziel ist inzwischen weitgehend erreicht. Dazu mussten insbesondere die Handelshemmnisse, die sich aus den unterschiedlichen nationalen Inverkehrbringens-Vorschriften für die Produkte ergaben, die am freien Warenverkehr teilnehmen sollen, abgebaut werden. Im Europäischen Parlament und Rat werden deshalb gemeinsame Vorschriften beschlossen. Dies wird als Harmonisierung der Rechtsvorschriften bezeichnet. Im Bereich des freien Warenverkehrs sind dies in der Regel Richtlinien, die dann noch in nationales Recht übernommen werden müssen. Diese Richtlinien sind heute auf Artikel 95 des EG Vertrages gestützt und enthalten formale Anforderungen wie auch Sicherheits- und Gesundheitsanforderungen.

Die EG besteht heute aus 25 Mitgliedstaaten:

Belgien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Italien, Lettland, Litauen, Luxemburg, Malta, Niederlanden, Österreich, Polen, Portugal, Schweden, Slowakei, Slowenien, Spanien, Tschechien, Ungarn, das Vereinigte Königreich und Zypern.

Zusammen mit den drei Staaten der Europäische Freihandelszone (EFTA):

Island, Liechtenstein und Norwegen

bilden die EG Staaten den europäischen Wirtschaftsraum (EWR), in dem hinsichtlich des freien Wahrenverkehrs die gleichen Vorschriften gelten.

Die Schweiz hat politisch gesehen eine Sonderrolle, kann aber faktisch im Bereich des Handels mit Maschinen, und somit auch mit Maschinensteuerungen den oben genannten Staaten gleichgestellt werden.

Eine genaue historische Zusammenfassung der Bildung der EG sowie eine Erläuterung der Sonderrolle der Schweiz befinden sich in Anhang I dieser Diplomarbeit.

Zum Inhalt dieses Kapitels siehe auch Quellen [12], [19] und [31].

4.2 Europäische Maschinenrichtlinie

Für dieses Kapitel wurden u. a. Informationen aus den Artikeln [9] und [10] verwendet.

Die ersten Binnenmarktregelungen für Maschinen enthält die Niederspannungsrichtlinie 73/23/EWG. Diese greifen aber für Maschinen im Wesentlichen nur im Bereich der so genannten Verbraucherprodukte. Deshalb hatte bis zum Inkrafttreten der Maschinenrichtlinie am 1. Januar 1993 jedes Land in Europa seine eigenen Vorschriften für das Inverkehrbringen von Maschinen. Zum Exportieren einer Maschine musste bis dahin erst einmal geprüft werden, welche technischen und formalen Anforderungen in dem entsprechenden Land galten, in dem diese Maschine in den Verkehr gebracht werden sollte. Dazu kamen unterschiedliche Prüfanforderungen in den verschiedenen Ländern. Selbst bei gleichen Prüfanforderungen gab es teilweise Probleme mit der Anerkennung der Zertifikate ausländischer Zertifizierer. Weiterhin waren für einige Maschinen von Staat zu Staat unterschiedliche Genehmigungsverfahren vorgeschrieben. Am Ende musste die gleiche Maschine ggf. für fünf verschiedene Länder in fünf verschiedenen Ausführungen gefertigt werden und dann vielleicht noch von fünf unterschiedlichen Zertifizierungsstellen begutachtet werden. Die fünf möglicherweise unterschiedlichen Genehmigungsverfahren dürfen dabei nicht vergessen werden. Dies alles bedeutete einen enormen Aufwand für die Industrie.

Dieses Kapitel des europäischen Warenverkehrs für Maschinen ist insbesondere mit Inkrafttreten der Maschinenrichtlinie 98/37/EG (MRL) vorbei. Für einzeln in Verkehr gebrachte Sicherheitsbauteile, die ebenfalls unter den Anwendungsbereich der MRL fallen endete es allerdings erst zum 1. Januar 2005. Die MRL ist heute Garant für einheitliche sicherheitstechnische Anforderungen an Maschinen und Sicherheitsbauteile im Binnenmarkt. Konformitätsbewertungs- und die Zertifizierungsverfahren wurden ebenfalls durch diese Richtlinie vereinheitlicht. Die letzte Übergangsfrist zur Anwendung der Richtlinie ist seit dem 1. Januar 1997 abgelaufen.

Die Anforderungen der MRL an das Inverkehrbringen von Maschinen, Sicherheitsbauteilen und Teilmaschinen sind in der Abbildung 1 schematisch dargestellt. Auf die einzelnen Schritte wird in den nachfolgenden Kapiteln näher eingegangen.

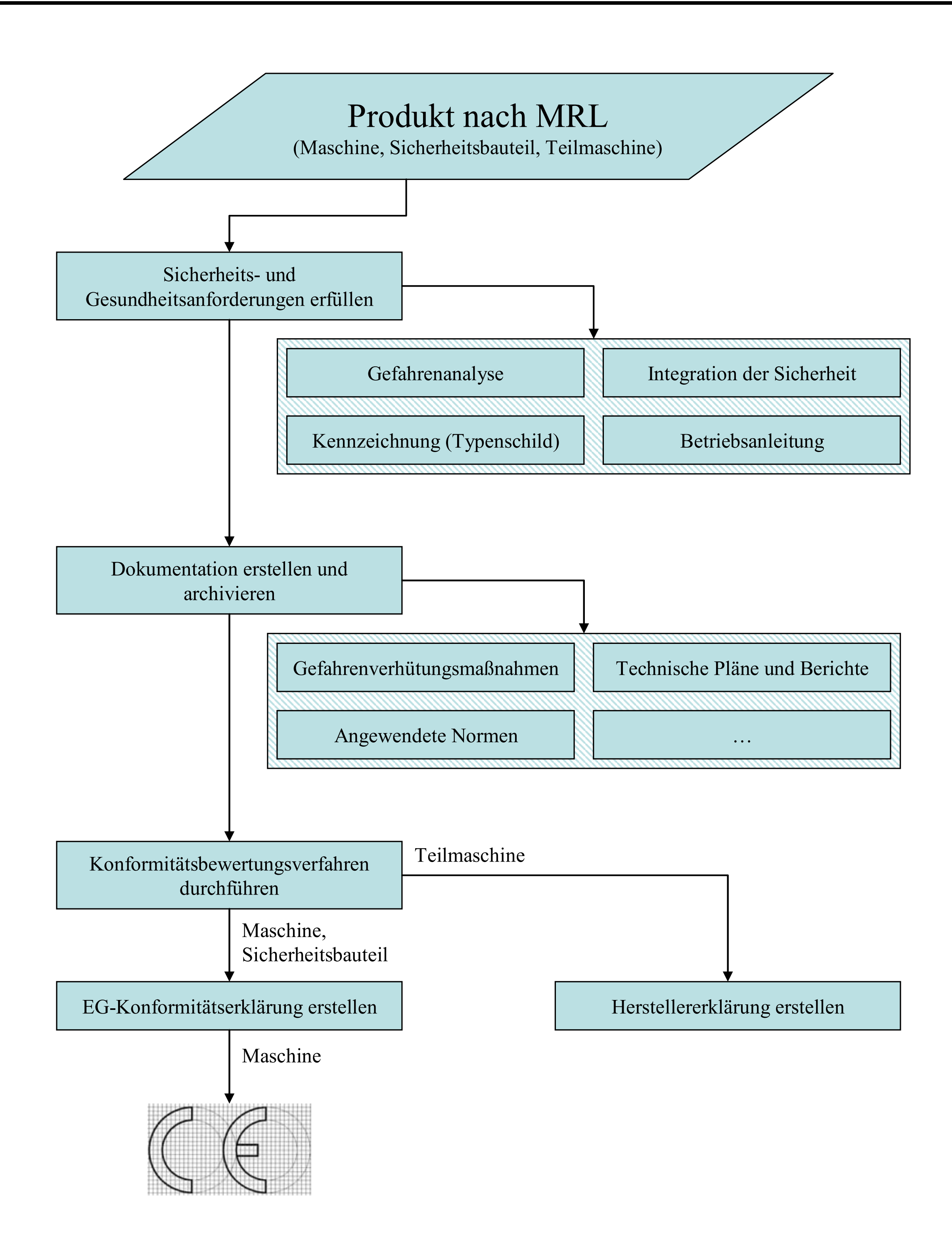


Abbildung 1: Anforderungen der Maschinenrichtlinie

Eine Neufassung der MRL, die Fehler und Unsicherheiten der jetzt gültigen Richtlinie bereinigt und für mehr Rechtssicherheit sorgen soll, wurde am 25. April 2006 von den Mitgliedstaaten verabschiedet und am 17. Mai 2006 ratifiziert. Diese „Neue Maschinenrichtlinie" trägt die Nummer 2006/42/EG. Sie übernimmt wesentliche Elemente der jetzt noch gültigen Richtlinie, so dass sich für den „normalen" Maschinenhersteller wenig ändern wird. Eine wesentliche Änderung ergibt sich jedoch im Bereich der Sicherheitsbauteile, die dem Anhang IV zugeordnet sind. Dies betrifft insbesondere

speicherprogrammierbare Steuerungen (SPS) mit Diagnosefunktionen wenn sie einzeln in Verkehr gebracht werden (s. u.). Die neue Maschinenrichtlinie tritt am 29. Dezember 2009 in Kraft.

Die Maschinenrichtlinie hat einen sehr großen Anwendungsbereich. Ihrem Namen zufolge erstreckt sie sich in erster Linie auf Maschinen, wobei Maschinenanlagen und auswechselbare Ausrüstungen den Maschinen gleichgestellt sind. Weiterhin erstreckt sich die Richtlinie auf Sicherheitsbauteile, sowie auf Lastaufnahmeeinrichtungen. Letztere haben einen besonderen Stand in der MRL, da sie in der jetzt gültigen Richtlinie zwar hinsichtlich der technischen Anforderungen geregelt werden, nicht aber unter den Anwendungsbereich der MRL fallen. Dieser Fehler ist in der neuen MRL beseitigt. Sie fallen jetzt unter den Anwendungsbereich.

Nach Artikel 1 Absatz 2 a) der Maschinenrichtlinie gilt als Maschine:

> *"Eine Gesamtheit von miteinander verbundenen Teilen oder Vorrichtungen, von denen mindestens eines beweglich ist, sowie ggf. von Betätigungsgeräten, Steuer- und Energiekreisen usw., die für eine bestimmte Anwendung, wie die Verarbeitung, die Behandlung, die Fortbewegung und die Aufbereitung eines Werkstoffes zusammengefügt sind". [26]*

Ein Gerät benötigt nach der Maschinenrichtlinie also weder eine Antriebsquelle, noch irgendwelche Betätigungsgeräte, Steuer- und Energiekreise um als Maschine zu gelten und damit unter die Richtlinie zu fallen. Die Richtlinie umfasst somit nicht nur funktionsfähige Maschinen, sondern auch so genannte Teilmaschinen, die erst noch, z.B. mit anderen Maschinen, zu einer funktionsfähigen Maschine zusammengeführt werden müssen.

Nicht Bestandteil der Maschinendefinition ist die Maschinensteuerung, wozu auch die SPS gehört. D. h. das Inverkehrbringen einer Steuerung fällt grundsätzlich – für sich allein genommen - nicht unter den Anwendungsbereich der MRL. Die MRL greift für eine Steuerung erst dann, wenn diese Bestandteil einer Maschine ist, da sie auch an die Steuerung einer Maschine technische Anforderungen stellt.

Der Anwendungsbereich der MRL erstreckt sich, wie bereits oben ausgeführt, nach Artikel 1 Absatz 2 allerdings auch auf „einzeln inverkehrgebrachte Sicherheitsbauteile“:

> *„Soweit es sich nicht um auswechselbare Ausrüstungen handelt, gelten im Sinne dieser Richtlinie als Sicherheitsbauteile jene Bauteile, die vom Hersteller oder seinem in der Gemeinschaft niedergelassenen Bevollmächtigten mit dem Verwendungszweck der Gewährleistung einer Sicherheitsfunktion in den Verkehr gebracht werden und deren Ausfall oder Fehlfunktion die Sicherheit oder die Gesundheit der Personen im Wirkbereich der Maschine gefährdet.“ [26]*

D. h. soweit die Steuerung ein Sicherheitsbauteil in diesen Sinne ist, ist die MRL auch auf das Inverkehrbringen dieser Steuerung anzuwenden. Insbesondere sind in diesem Zusammenhang die in Anhang IV B der MRL aufgeführten Sicherheitsbauteile nach Nr. 2 zu beachten:

> *„Logikeinheiten zur Aufrechterhaltung der Sicherheitsfunktionen von Zweihandschaltungen“ [26]*

Diese Sicherheitsbauteile unterliegen nach der MRL einem besonderen Konformitätsbewertungsverfahren, in dem eine benannte Stelle eingeschaltet werden muss.

Mit der neuen Maschinenrichtlinie 2006/42/EG, die am 29. Dezember 2009 in Kraft treten wird, wird dieser Punkt wesentlich ausgeweitet. Nach Anhang IV, Nr. 21 der neuen Maschinenrichtlinie werden jetzt erfasst:

„Logikeinheiten für Sicherheitsfunktionen.“ [27]

Damit fällt zukünftig jede einzeln inverkehrgebrachte SPS mit Diagnosefunktionen unter das besondere Konformitätsbewertungsverfahren (siehe Kapitel 4.2.1.3) für Anhang IV Sicherheitsbauteile. Das sind neben der in Kapitel 6.1.4 beschriebenen SSPS auch solche SPS die mit den in dieser Diplomarbeit beschriebenen Funktionsbausteinen erweitert wurden. D.h. dieses Konformitätsbewertungsverfahren wird nicht auf SPSen mit Diagnosefunktion angewendet, die zusammen mit einer Maschine in Verkehr gebracht werden. (siehe Abbildung 2)

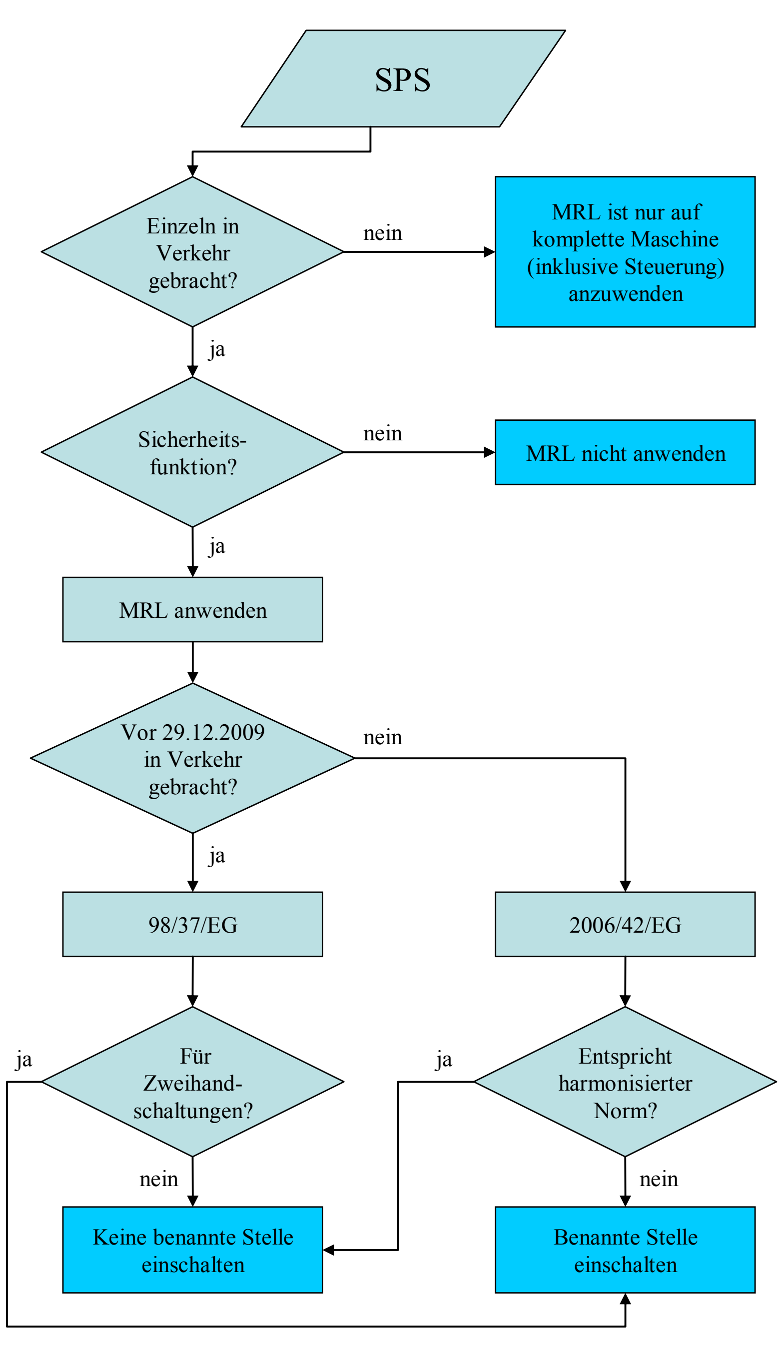


Abbildung 2: Konfomitätsbewertung nach Maschinenrichtlinie für SPS

Die MRL enthält für das Inverkehrbringen von Maschinen und Sicherheitsbauteilen formale wie auch technische Anforderungen, die nachfolgend beschrieben werden.

4.2.1 Formale Anforderungen nach 98/37/EG[1]

4.2.1.1 Kennzeichnung

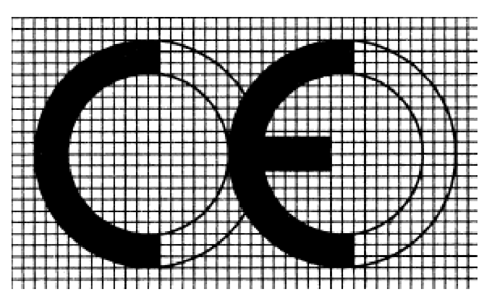
Die MRL enthält insbesondere zwei Kennzeichnungsvorgaben für Maschinen und Sicherheitsbauteile. Zum einen ist es die CE-Kennzeichnung, die nur bei – für sich alleine funktionsfähigen - Maschinen und nicht bei Sicherheitsbauteilen sowie Teilmaschinen nach Artikel 4 Abs. 2 MRL anzubringen ist. Zum anderen sind es Vorgaben für den Inhalt des anzubringenden Typenschildes.

Zur CE-Kennzeichnung enthält Artikel 8 Absatz 1 folgendes:

„Der Hersteller oder sein in der Gemeinschaft niedergelassener Bevollmächtigter muss, um die Übereinstimmung der Maschinen und Sicherheitsbauteile mit den Bestimmungen dieser Richtlinie zu bescheinigen, für jede hergestellte Maschine bzw. jedes hergestellte Sicherheitsbauteil eine EG-Konformitätserklärung gemäß Anhang II Buchstabe A bzw. Buchstabe C ausstellen.
Ferner muss der Hersteller oder sein in der Gemeinschaft niedergelassener Bevollmächtigter — nur auf Maschinen — die genannte CE-Kennzeichnung anbringen." [26]

Die Form und Größe der CE-Kennzeichnung ergibt sich aus Anhang III der MRL:

„CE-KONFORMITÄTSKENNZEICHNUNG
— Die CE-Konformitätskennzeichnung besteht aus den Buchstaben „CE" mit folgendem Schriftbild:



— […]" [26]

Zu beachten ist, dass die CE Kennzeichnung keine freiwillig anzubringende Kennzeichnung ist. Sie ist nach Artikel 8 Absatz 1 auf Maschinen immer anzubringen. Eine falsche Kennzeichnung mit dem CE-Kennzeichen ist nach § 6 Absatz 1 Geräte- und Produktsicherheitsgesetz (GPSG) verboten:

„Es ist verboten, ein Produkt in den Verkehr zu bringen, wenn dieses, seine Verpackung oder ihm beigefügte Unterlagen mit der CE-Kennzeichung versehen sind, ohne dass die Rechtsverordnungen nach § 3 oder andere Rechtsvorschriften dies vorsehen und die Voraussetzungen der Absätze 2 bis 5 eingehalten sind." [28]

Danach dürfen Sicherheitsbauteile aufgrund des Artikel 8 Absatz 1 MRL (s. o.) grundsätzlich nicht mit dem CE-Kennzeichen versehen werden. Allerdings fallen diverse Sicherheitsbauteile auch unter den Anwendungsbereich anderer Richtlinien, wie die Niederspannungsrichtlinie oder die Richtlinie

[1] Die formalen Anforderungen für das in Verkehr bringen von SPSen ändern sich zum Teil mit der neuen Maschinenrichtlinie 2006/42/EG (siehe Abbildung 2). Auf diese Änderungen, die erst zum 29.12.2009 in Kraft treten, wird in dieser Arbeit, wo erforderlich, hingewiesen, sie werden aber nicht ausführlich behandelt, da sie noch kein geltendes Recht sind.

über die elektromagnetische Verträglichkeit, die die CE-Kennzeichnung wiederum verlangen. Viele Sicherheitsbauteile müssen aus diesem Grund mit einer CE-Kennzeichnung versehen werden. Die neue MRL wird diesen verwirrenden Zustand ändern, indem sie auch für Sicherheitsbauteile eine CE-Kennzeichnung vorschreibt.

4.2.1.2 Dokumentation nach MRL

Die vom Hersteller zu erstellenden Unterlagen (Dokumentation) unterscheiden sich danach, ob die Maschine oder das Sicherheitsbauteil im Anhang IV MRL gelistet ist oder nicht. Nach Art. 8 Abs. 2 a muss der Hersteller bei Maschinen und Sicherheitsbauteilen, die nicht im Anhang IV gelistet sind, die Unterlagen nach Anhang V Nr. 3 MRL zusammenstellen. Im anderen Fall sind dies nach Art. 8 Abs. 2 b bzw. c die Unterlagen nach Anhang VI. Eine Übersicht über den Umfang und die Bedeutung der Unterlagen befindet sich im Anhang I der Diplomarbeit.

Aus diesem Grund sind für das in Verkehr bringen einer SPS drei Fälle zu unterscheiden (siehe hierzu auch Abbildung 2), die auch auf eine mit den in dieser Arbeit beschriebenen Tests sicherheitstechnisch erweiterte SPS anzuwenden sind:

- Für die Standard-SPS ohne Diagnosefunktion bzw. eine zusammen mit einer Maschine in Verkehr gebrachte SPS mit und ohne Diagnosefunktion bedeutet dies, dass die SPS-Dokumentation Bestandteil der Dokumentation der gesamten Maschine ist.
- Für die einzeln in Verkehr gebrachte SPS mit Diagnosefunktion, die nicht für Zweihandschaltungen gedacht ist, ist vom Hersteller eine Dokumentation nach Anhang V MRL zu erstellen.
- Für die einzeln in Verkehr gebrachte SPS mit Diagnosefunktion, die für Zweihandschaltungen gedacht ist, ist vom Hersteller eine Dokumentation nach Anhang VI MRL zu erstellen.

4.2.1.3 Konformitätsbewertung

Die Konformitätsbewertungsverfahren sind in der EU standardisiert. Sie sind grundsätzlich nach dem Beschluss des Rates 93/465/EWG über die in den Binnenmarktrichtlinien zu verwendenden Konformitätsbewertungsmodule [23] geregelt. Das so genannte Modulpapier kennt folgende acht Konformitätsbewertungsmodule:

- A - interne Fertigungskontrolle
- B - EG-Baumusterprüfung
- C - Konformität mit der Bauart
- D - Qualitätssicherung Produktion
- E - Qualitätssicherung Produkt
- F - Prüfung der Produkte
- G - Einzelprüfung
- H - umfassende Qualitätssicherung

Die Konformitätsbewertung ist nach DIN EN 45020 die systematische Untersuchung, in wieweit ein Produkt, ein Prozess oder eine Dienstleistung festgelegte Anforderungen erfüllt. Dies bedeutet, bezogen auf die MRL, dass in einem Verfahren - dem Konformitätsbewertungsverfahren - festgestellt wird, ob ein unter die Richtlinie fallendes Produkt, z. B. eine mit den in dieser Arbeit beschriebenen Tests erweiterte SPS, die formalen, wie auch die Sicherheits- und Gesundheitsanforderungen der MRL erfüllt. Nur wenn dies gewährleistet ist, kann die Maschine oder das Sicherheitsbauteil am freien Warenverkehr im europäischen Binnenmarkt teilnehmen.

Die im Rahmen der Konformitätsbewertung zu prüfenden Anforderungen der MRL können in folgende Punkte aufteilt werden:

- Sicherheit- und Gesundheitsanforderungen (Anhang I MRL, siehe Kapitel 4.2.2)
  - Gefahrenanalyse
  - Integration der Sicherheit
  - Kennzeichnung
  - Betriebsanleitung
- Ggf. Einschaltung einer benannten Stelle (siehe Anhang I der Diplomarbeit)
  - Erstellung der Unterlagen nach Anhang VI MRL
  - Baumusterprüfung (Art. 8 Abs. 2 in Verbindung mit Anhang VI MRL)
  - Prüfung der Unterlagen (Art. 8 Abs. 2 in Verbindung mit Anhang VI MRL)
  - Aufbewahrung der Unterlagen (Art. 8 Abs. 2 in Verbindung mit Anhang VI MRL)
- Dokumentation (Anhang V MRL, siehe Anhang I der Diplomarbeit)
- Bescheinigungen
  - EG-Konformitätserklärung (Anhang II A oder C MRL, siehe Kapitel 4.2.1.4)

  oder

  - Herstellererklärung (Anhang II B MRL)
- CE-Kennzeichnung (Anhang III MRL, siehe Kapitel 4.2.1.1)

Nicht vernachlässigt werden darf allerdings, dass in der Regel neben der MRL noch andere Binnenmarktrichtlinien anzuwenden sind, wie z.B. die Niederspannungsrichtlinie, die ATEX-Richtlinie [24], die EMV-Richtlinie [22] und die Druckgeräterichtlinie [25], die eigene Konformitätsbewertungsverfahren vorschreiben und die ggf. zusätzlich anzuwenden sind. Auch Anforderungen des Umweltschutzes sind ggf. für Maschinen einzuhalten.

Nachfolgend beschränken sich die Ausführungen jedoch auf die Bestimmungen der MRL und der EMV-Richtlinie, die bei Maschinensteuerungen, wie die SPS in der Regel anzuwenden sind. Die im Maschinenbau ansonsten regelmäßig zum Tragen kommende Niederspannungsrichtlinie kommt auf

Grund ihrer Spannungsgrenzen (50 bis 1000 V Wechselstrom bzw. 75 bis 1500 V Gleichstrom) bei den hier behandelten SPSen eher selten zur Anwendung. (siehe auch Kapitel 4.2.3)

Die MRL schreibt grundsätzlich das Konformitätsbewertungsverfahren nach „Modul A" (interne Fertigungskontrolle) vor, d. h. der Maschinen- bzw. Sicherheitsbauteilehersteller bewertet die Konformität allein ohne die Einschaltung Dritter. Für einen kleineren Teil der Maschinen und Sicherheitsbauteile, die in Anhang IV MRL aufgeführt sind, ist das Modul B (Baumusterprüfung) anzuwenden. D. h. eine so genannte benannte Stelle muss eingeschaltet werden um ein Baumuster hinsichtlich der Übereinstimmung mit der MRL zu prüfen. Das Modul B ist in der MRL zum Modulpapier hin ergänzt worden, so dass der Hersteller bei der Einhaltung harmonisierter Normen anstelle der Baumusterprüfung auch die benannte Stelle im Rahmen des Verfahrens „Prüfung der Unterlagen" oder „Aufbewahrung der Unterlagen" einschalten kann.

Aus diesem Grund sind für das in Verkehr bringen einer SPS analog zu Kapitel 4.2.1.2 drei Fälle zu unterscheiden, die auch auf eine mit den in dieser Arbeit beschriebenen Tests sicherheitstechnisch erweiterte SPS anzuwenden sind:

- Für die Standard-SPS ohne Diagnosefunktion bzw. eine zusammen mit einer Maschine in Verkehr gebrachte SPS mit und ohne Diagnosefunktion bedeutet dies, dass die Konformität der SPS im Rahmen der Konformitätsbewertung der gesamten Maschine geprüft wird.
- Für die einzeln in Verkehr gebrachte SPS mit Diagnosefunktion, die nicht für Zweihandschaltungen gedacht ist, ist vom Hersteller eine gesonderte Konformitätsbewertung nach Artikel 8 Abs. 2 a) in Verbindung mit Artikel 8 Abs. 5 MRL durchzuführen.
- Für die einzeln in Verkehr gebrachte SPS mit Diagnosefunktion, die für Zweihandschaltungen gedacht ist, ist vom Hersteller eine Konformitätsbewertung nach Artikel 8 Abs. 2 b) bzw. c) in Verbindung mit Artikel 8 Abs. 5 MRL durchzuführen.

4.2.1.4 EG-Konformitätserklärung

Als EG-Konformitätserklärung wird nach Anhang V Nr. 1 das Verfahren bezeichnet, bei dem der Hersteller oder ein in der Gemeinschaft niedergelassener Bevollmächtigter erklärt, dass die in den Verkehr gebrachte Maschine allen einschlägigen grundlegenden Sicherheits- und Gesundheitsanforderungen entspricht. Die Unterzeichnung der EG-Konformitätserklärung berechtigt erst den Hersteller das CE-Kennzeichen (nur bei Maschinen, nicht bei Sicherheitsbauteilen – siehe Kapitel 4.2.1.1) anzubringen. Der Inhalt der EG-Konformitätserklärung ergibt sich aus Anhang II A MRL für Maschinen und aus Anhang II C MRL für Sicherheitsbauteile. Eine bestimmte Form ist nicht vorgeschrieben.

Die EG-Konformitätserklärung für einzeln in Verkehr gebrachte Sicherheitsbauteile, wie z. B. eine SPS mit Diagnosefunktionen, muss nach Anhang II C MRL folgende Angaben enthalten:

- *„Name und Anschrift des Herstellers oder seines in der Gemeinschaft niedergelassenen Bevollmächtigten;*
- *Beschreibung des Sicherheitsbauteils;*
- *Sicherheitsfunktion des Sicherheitsbauteils, falls diese aus der Beschreibung nicht klar ersichtlich ist;*
- *gegebenenfalls Name und Anschrift der gemeldeten Stelle und Nummer der EG-Baumusterbescheinigung;*
- *gegebenenfalls Name und Anschrift der gemeldeten Stelle, der die Unterlagen gemäß Artikel 8 Absatz 2 Buchstabe c) erster Gedankenstrich übermittelt worden sind;*
- *gegebenenfalls Name und Anschrift der gemeldeten Stelle, die die Überprüfung gemäß Artikel 8 Absatz 2 Buchstabe c) zweiter Gedankenstrich vorgenommen hat;*
- *gegebenenfalls die Fundstellen der harmonisierten Normen;*
- *gegebenenfalls die Fundstellen der nationalen Normen und technischen Spezifikationen, die verwendet wurden;*
- *Angaben zum Unterzeichner, der bevollmächtigt ist, die Erklärung für den Hersteller oder seinen in der Gemeinschaft niedergelassenen Bevollmächtigten rechtsverbindlich zu unterzeichnen." [26]*

Die Erklärung ist eine Erklärung des Herstellers. D.h. unterzeichnet werden kann die EG-Konformitätserklärung vom Geschäftsführer oder von einem dazu beauftragten Mitarbeiter, der dann mit dem Zusatz i. V. oder i. A. unterschreibt.

Nachfolgend ist ein Muster einer EG-Konfirmitätserklärung für Sicherheitsbauteile abgedruckt:

MUSTER

EG-KONFORMITÄTSERKLÄRUNG FÜR EINZELN IN VERKEHR GEBRACHTE SICHERHEITSBAUTEILE*)

(EG-Maschinen-Richtlinie 98/37/EG)

Hiermit erklärt der Hersteller:..

....

(Name, Rechtsform, Anschrift)

daß der / die / das

....

(eindeutige Beschreibung des Sicherheitsbauteils: Fabrikat, Typ, Seriennummer, etc.)

- konform ist mit den einschlägigen Bestimmungen der EG-Maschinen-Richtlinie (98/37/EG), einschließlich ihrer zum Zeitpunkt der Erklärung geltenden Änderungen
- konform ist mit den einschlägigen Bestimmungen folgender weiterer EG-Richtlinien und ihrer zum Zeitpunkt der Erklärung geltenden Änderungen**) :
  - ⇒ ....
  - ⇒ ....
- folgende harmonisierte Normen (oder Teile hieraus) angewandt wurden**):
  (Ausgabedatum mit angeben)
  - ⇒ ....
  - ⇒ ....
- folgende nationale Normen und technische Spezifikationen angewandt wurden**):
  (mit Ausgabedatum angeben)
  - ⇒ ....
  - ⇒ ....
- folgende gemeldete Stelle eingeschaltet wurde *(Name und Anschrift)* wegen**):
  (zutreffendes bitte ankreuzen)

  ....

  ☐ EG-Baumusterprüfung Nr.: ....

  oder

  ☐ Übersendung der Unterlagen gemäß Artikel 8(2)c) erster Gedankenstrich

  oder

  ☐ Prüfung der Unterlagen gemäß Artikel 8(2)c) zweiter Gedankenstrich

Ort, Datum: ....

---

*) *Diese Erklärung ist in derselben Sprache wie die Originalbetriebsanleitung abzufassen (siehe Anhang I, Abschnitt 1.7.4 Buchstabe b)), und zwar - mit Ausnahme der handgeschriebenen Unterschrift- maschinenschriftlich oder in Druckbuchstaben. Ihr muß eine Übersetzung in einer der Sprachen des Verwenderlandes beigefügt sein. Für diese Übersetzung gelten die gleichen Bedingungen wie für die Betriebsanleitung.*

**) *Nur angeben wenn zutreffend*

4.2.2 Technische Anforderungen

4.2.2.1 Anhang I Maschinenrichtlinie

Die Verpflichtung des Herstellers zur Einhaltung bestimmter Sicherheits- und Gesundheitsanforderungen ergibt sich aus Artikel 3 der Maschinenrichtlinie. Hier wird bestimmt, dass Maschinen und Sicherheitsbauteile die in Anhang I der Maschinenrichtlinie aufgeführten Sicherheits- und Gesundheitsanforderungen einhalten müssen. Hie finden sich auch konkrete Anforderungen an Steuerungen, wie in Kapitel 5.1.1 näher erläutert wird. Ziel ist nach Artikel 2 Abs. 1 die Sicherheit und Gesundheit von Personen, Haustieren und Gütern durch Maschinen nicht zu gefährden.

Nach der harmonisierten Norm DIN EN 12100-1 wird unter der Sicherheit einer Maschine verstanden:

> *„3.4. Sicherheit einer Maschine*
> *Die Fähigkeit einer Maschine ihre Funktion(en) durchzuführen und transportiert, aufgebaut, eingerichtet, instandgehalten, abgebaut und entsorgt zu werden unter den Bedingungen der bestimmungsgemäßen Verwendung wie sie vom Hersteller in der Betriebsanleitung festgelegt ist (und auf die in einigen Fällen für bestimmte Zeitabschnitte auch in der Betriebsanweisung hingewiesen ist), ohne dass dadurch Verletzungen oder Gesundheitsschädigungen verursacht werden.“*

Anhang I MRL enthält teilweise sehr konkrete, teilweise aber auch sehr abstrakte Anforderungen, die allerdings immer absolut sind. Nicht vergessen werden dürfen in diesem Zusammenhang die Vorbemerkungen zum Anhang I. Aus den Vorbemerkungen wird klar, wie die absoluten Anforderungen des Anhangs zu lesen sind. Hier steht z. B., dass die Anforderungen nur in soweit erfüllt werden müssen, wie es der Stand der Technik es ermöglicht. Der Stand der Technik wird in den Rechtsvorschriften nach MRL gar nicht und in der Fachliteratur unterschiedlich definiert. In der Regel wird darunter das zum Zeitpunkt des in Verkehr bringen bestmögliche angewandte und zugängliche Verfahren verstanden, welches auch wirtschaftlich durchführbar ist. Eine passende Legaldefinition, die jedoch nicht auf die wirtschaftlichen Aspekte abhebt, findet sich jedoch im Paragraph 3 Absatz 10 der Gefahrstoffverordnung:

> *„Der „Stand der Technik“ ist der Entwicklungsstand fortschrittlicher Verfahren, Einrichtungen oder Betriebsweisen, der die praktische Eignung einer Maßnahme zum Schutz der Gesundheit und zur Sicherheit der Beschäftigten gesichert erscheinen lässt. Bei der Bestimmung des Standes der Technik sind insbesondere vergleichbare Verfahren, Einrichtungen oder Betriebsweisen heranzuziehen, die mit Erfolg in der Praxis erprobt worden sind. […]“ [29]*

Weitere wichtige Punkte der Vorbemerkungen sind die Verpflichtung zur Erstellung einer Gefahrenanalyse und zur Einhaltung der Grundsätze für das Sicherheitskonzept. Dies wird auch als Drei-Stufen-Methode bezeichnet, wie in Abbildung 3 darstellt.

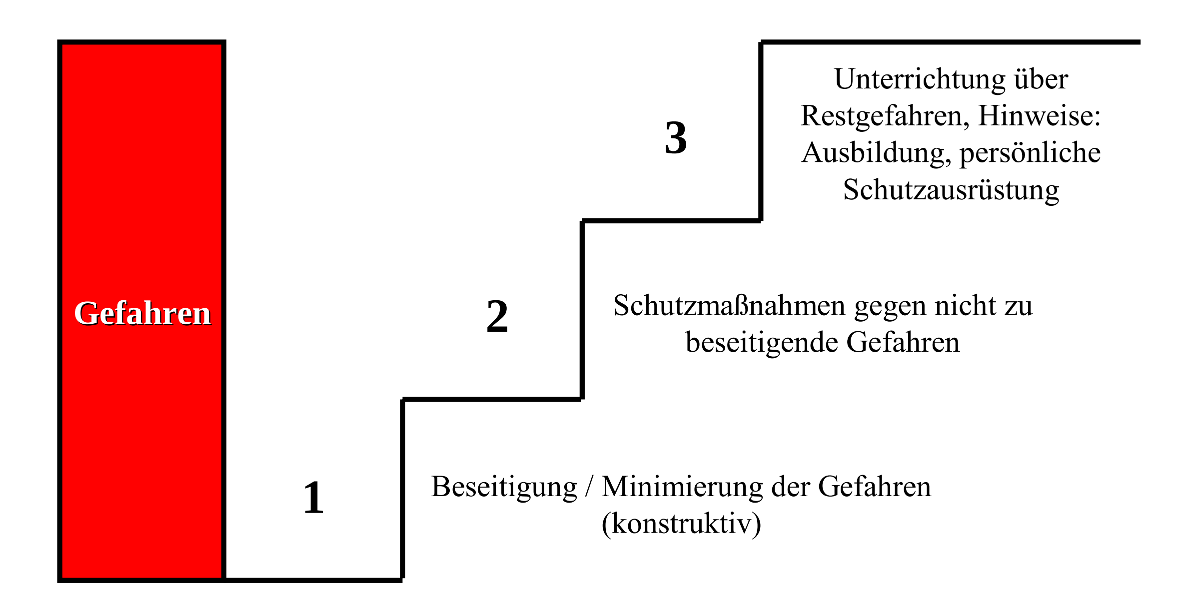


Abbildung 3: Grundsätze des Sicherheitskonzepts (Drei-Stufen-Methode)

Steuerungstechnische Maßnahmen sind der Stufe zwei zuzuordnen, da sie keine Gefahren beseitigen oder minimieren, sondern lediglich Schutzmaßnahmen darstellen. Insofern sind sicherheitstechnische Maßnahmen, die mit Hilfe einer SPS realisiert werden, Sekundärmaßnahmen und deshalb der Stufe zwei zuzuordnen.

Die Art der Erfüllung der Sicherheits- und Gesundheitsanforderungen ist dem Hersteller im Rahmen des Standes der Technik freigestellt. Ihm obliegt es im Streitfall mit der Behörde die Richtlinienkonformität seines Produktes nachzuweisen (Beweislast des Herstellers). Zu einer Umkehrung der Beweislast kommt es bei der Anwendung bestimmter harmonisierter Normen. (Siehe hierzu das Kapitel 4.2.2.2)

Der Anhang I MRL ist in 6 Abschnitte unterteilt. Sie enthalten grundlegende Sicherheits- und Gesundheitsanforderungen an:

- alle Maschinen
- bestimmte Maschinengattungen
    - Nahrungsmittelmaschinen
    - In der Hand gehaltene bzw. von Hand geführte Maschinen
    - Maschinen zur Bearbeitung von Holz und gleichartigen Werkstoffen
- bewegliche Maschinen
- Maschinen zum Heben
- Maschinen für den Untertagebau
- Maschinen zum Heben oder Fortbewegen von Personen

4.2.2.2 Harmonisierte Normen

Zur Zeit gibt es in den Binnenmarktrichtlinien keine Legaldefinition für den Begriff „harmonisierte Norm“, so dass die Aussagen im Binnenmarktleitfaden hier hilfreich sind. Hierbei handelt es sich allerdings nicht um eine Rechtsvorschrift, sondern um ein zwischen der europäischen Kommission und den Mitgliedstaaten vereinbartes Interpretationspapier. Maßgebend im Einzelfall ist in erster Linie die jeweils einschlägige Rechtsvorschrift.

Nach dem Binnenmarktleitfaden wird unter einer harmonisierten Norm verstanden:

- *„[…]europäische Normen, die von europäischen Normungsorganisationen aufgrund eines von der Kommission nach Anhörung der Mitgliedstaaten erteilten Auftrags gemäß den allgemeinen Leitlinien erarbeitet wurden, die zwischen der Kommission und den europäischen Normungsorganisationen vereinbart wurden.*
- *Als harmonisierte Normen im Sinne des neuen Konzepts werden die europäischen Normen angesehen, die europäische Normungsorganisationen der Kommission formell vorlegen und die in deren Auftrag erarbeitet oder ermittelt wurden.“ [3]*

Mit der neuen Maschinenrichtlinie 2006/42/EG wird diese rechtliche Lücke allerdings geschlossen. Danach wird die „harmonisierte Norm“ in Artikel 2 wie folgt definiert:

> *„Ferner Bezeichnet der Ausdruck*
> *[…]*
> *l) „harmonisierte Norm“ eine nicht verbindliche technische Spezifikation, die von einer europäischen Normenorganisation, nämlich dem Europäischen Komitee für Normung (CEN), dem Europäischen Komitee für Elektrotechnische Normung (Cenelec) oder dem Europäischen Institut für Telekommunikationsnormen (ETSI), aufgrund eines Auftrags der Kommission nach den in der Richtlinie 98/34/EG des Europäischen Parlaments und des Rates vom 22. Juni 1998 über ein Informationsverfahren auf dem Gebiet der Normen und technischen Vorschriften und der Vorschriften für die Dienste der Informationsgesellschaft festgelegten Verfahren angenommen wurde.“ [27]*

Der Maschinen- und Sicherheitsbauteilehersteller kann bei Anwendung einer solchen harmonisierten Norm allerdings nicht davon ausgehen, dass er die Anforderungen der Maschinenrichtlinie erfüllt. Dazu muss die harmonisierte Norm erst im Amtsblatt der Europäischen Gemeinschaften veröffentlicht worden sein. Dies führt zur so genannten „Konformitätsvermutung“ – Umkehr der Beweislast zu Gunsten des Herstellers – nach Artikel 5 Absatz 2 MRL:

> *„Entspricht eine nationale Norm in Umsetzung einer harmonisierten Norm, deren Fundstelle im Amtsblatt der Europäischen Gemeinschaften veröffentlicht worden ist, einer oder mehreren grundlegenden Sicherheitsanforderungen, wird bei nach dieser Norm hergestellten Maschinen oder Sicherheitsbauteilen davon ausgegangen, dass sie den betreffenden grundlegenden Anforderungen genügen.“ [26]*

Diese Festlegung wird inhaltlich von der neuen MRL [27] in Artikel 7 Absatz 2 übernommen.

Die Anwendung harmonisierter Normen ist nicht vorgeschrieben, sie bietet allerdings durch die o. a. Konformitätsvermutung eine besondere Rechtssicherheit.

Der Gesamtzusammenhang zwischen der harmonisierten Norm und der Konformitätsvermutung ist in der folgenden Abbildung 4 dargestellt.

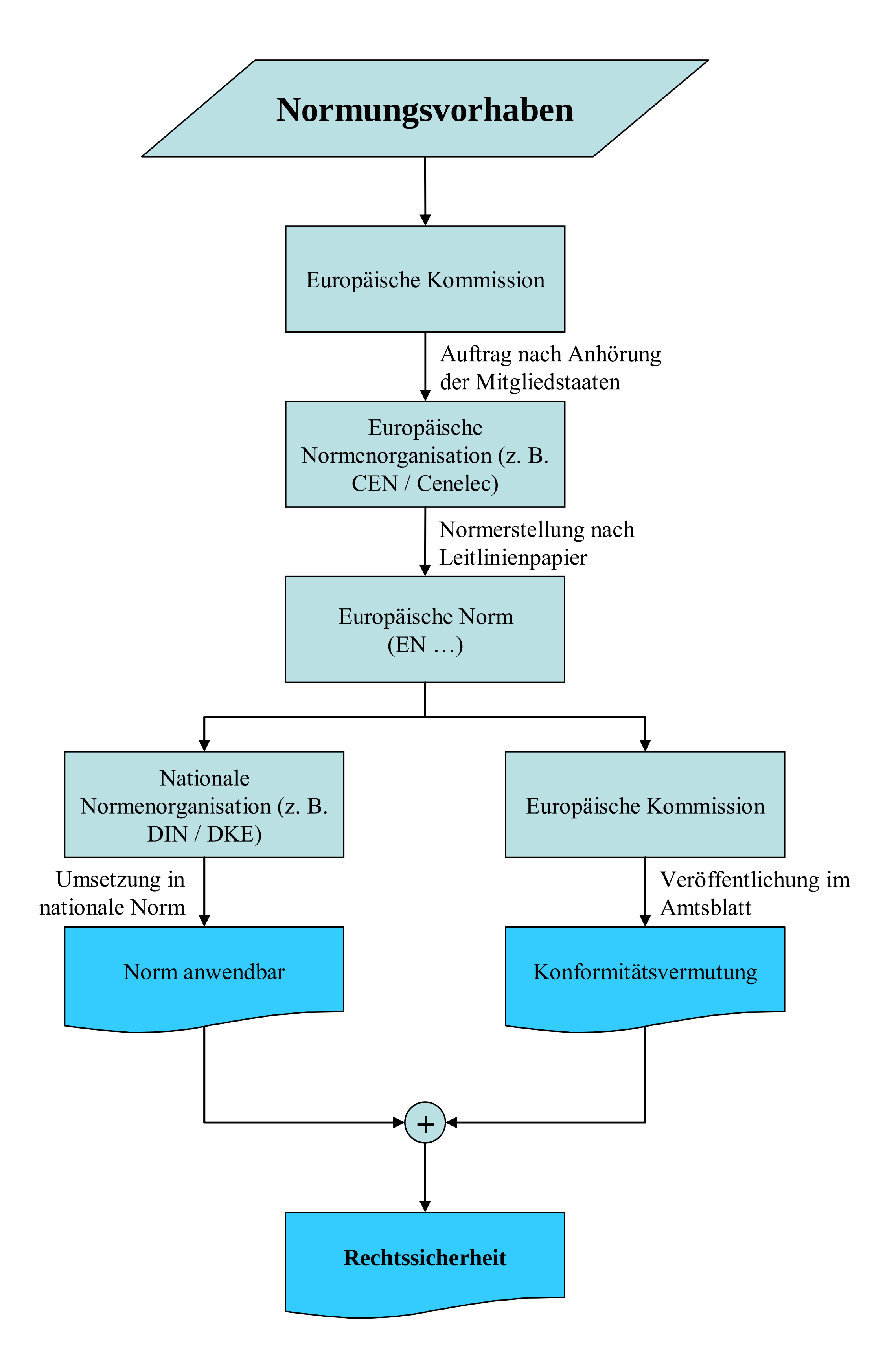


Abbildung 4: Europäische Harmonisierte Normen

4.2.3 Spezialrichtlinien

Nach Artikel 1 Abs. 4 MRL müssen ggf. neben der Maschinenrichtlinie auch andere Richtlinie beachtet werden:

> *„Werden die in dieser Richtlinie genannten Gefahren, die von einer Maschine oder einem Sicherheitsbauteil ausgehen, ganz oder teilweise von anderen besonderen Gemeinschaftsrichtlinien erfasst, so gilt diese Richtlinie für diese Maschine oder dieses Sicherheitsbauteil und diese Gefahren nicht bzw. findet sie auf diese ab Inkrafttreten der besonderen Richtlinie keine Anwendung mehr.“ [26]*

Solche Richtlinien können für Steuerungen sein:

- Niederspannungsrichtlinie [20] für die elektrischen Gefahren im Sinne von Anhang I, Nr. 1.5.1 MRL [26]:

  > *„Eine elektrisch angetriebene Maschine muss so konzipiert, gebaut und ausgerüstet sein, dass alle Gefahren aufgrund von Elektrizität vermieden werden oder vermieden werden können. Soweit die Maschine unter die spezifischen Rechtsvorschriften betreffend elektrische Betriebsmittel zur Verwendung innerhalb bestimmter Spannungsgrenzen fällt, sind diese anzuwenden.“ [20]*

  Wegen der Spannungsgrenzen (50 bis 1000 V Wechselstrom bzw. 75 bis 1500 V Gleichstrom) der Niederspannungsrichtlinie fallen die hier behandelten SPSen grundsätzlich nicht in deren Anwendungsbereich. (siehe auch Kapitel 4.2.1.3) Soweit die SPS elektrische Anschlüsse aufweist, die innerhalb der o. a. Spannungsgrenzen liegen, regelt die Niederspannungsrichtlinie die Vermeidung der hiervon ausgehenden elektrischen Gefahren.

- ATEX Richtlinie [24] für die Explosionsgefahren im Sinne von Anhang I, Nr. 1.5.7 MRL [26]:

  > *„[…]*
  > *- Die zu diesen Maschinen gehörenden elektrischen Betriebsmittel müssen hinsichtlich der Explosionsgefahr den geltenden Einzelrichtlinien entsprechen.“ [24]*

  Die ATEX Richtlinie kommt nur in Sonderfällen für die SPS zur Anwendung, nämlich dann, wenn diese direkt in der explosionsfähigen Atmosphäre eingesetzt wird. Ziel der ATEX-Richtlinie ist es im Bereich der SPSen, diese so zu bauen, dass diese nicht zur Zündquelle für eine explosionsfähige Atmosphäre werden.

- EMV Richtlinie [22] für die Gefahren durch Strahlung im Sinne von Anhang I, Nr. 1.5.10 MRL [26]:

  > *„Die Maschine muss so konzipiert und gebaut sein, dass jegliche Emission von Strahlung durch die Maschine auf das für ihr Funktionieren notwendige Maß beschränkt wird und eine Einwirkung auf die gefährdeten Personen vollständig unterbunden oder auf ein ungefährliches Maß begrenzt wird.“ [22]*

  Die EMV-Richtlinie [22] ist bei SPSen regelmäßig neben der Maschinenrichtlinie anzuwenden. Ihre grundlegenden Anforderungen werden in Kapitel 5.1.2 dargestellt.

  Die neue EMV-Richtlinie 2004/108/EG ist zur Zeit noch kein geltendes Recht. Hinsichtlich der Schutzziele im Bereich der SPSen ergeben sich jedoch keine Veränderungen. Die neue EMV-Richtlinie kann ab dem 20.7.2007 angewendet werden. Es besteht eine zwei-jährige

Übergangsfrist bis zum 20.7.2009, während der die alte Richtlinie weiterhin angewendet werden kann.

4.3 Nationales Recht als Umsetzung des EU-Rechts

Die europäischen Richtlinien wenden sich an die Mitgliedsstaaten und müssen von diesen in nationales Recht umgesetzt werden. Erst das nationale Recht ist für den Hersteller / Inverkehrbringer verbindlich. Im Bereich der Binnenmarktrichtlinien nach Artikel 95 EG-Vertrag [19] (freier Warenverkehr) haben die Mitgliedstaaten allerdings keinen „Spielraum" bei der Umsetzung. Sie müssen diese Richtlinien inhaltlich eins-zu-eins in nationales Recht übernehmen. In sofern muss auf die Inhalte der nationalen Rechtsvorschriften hier nicht weiter eingegangen werden.

Die nachfolgende Tabelle (Abbildung 5) enthält eine Übersicht über die für Maschinen und Sicherheitsbauteile wichtigsten europäischen Binnenmarktrichtlinien und ihre Umsetzungen in das nationale deutsche Recht:

EG-Richtlinie		**Umsetzung in nationales deutsches Recht**
73/23/EWG	Niederspannungsrichtlinie [20]	Geräte- und Produktsicherheitsgesetz (GPSG), in Verbindung mit der 1. Verordnung zum Geräte- und Produktsicherheitsgesetz (1. GPSGV)
89/336/EWG	EMV-Richtlinie [22]	Gesetz über die elektromagnetische Verträglichkeit (EMVG)
94/9/EG	Ex-Schutz Richtlinie [24]	Geräte- und Produktsicherheitsgesetz (GPSG), in Verbindung mit der 11. Verordnung zum Geräte- und Produktsicherheitsgesetz (11. GPSGV)
87/404/EWG	Richtlinie einfache Druckbehälter [21]	Geräte- und Produktsicherheitsgesetz (GPSG), in Verbindung mit der 6. Verordnung zum Geräte- und Produktsicherheitsgesetz (6. GPSGV)
97/23/EG	Druckgeräte-Richtlinie [25]	Geräte- und Produktsicherheitsgesetz (GPSG), in Verbindung mit der 14. Verordnung zum Geräte- und Produktsicherheitsgesetz (14. GPSGV)
98/37/EG	Maschinenrichtlinie [26]	Geräte- und Produktsicherheitsgesetz (GPSG), in Verbindung mit der 9. Verordnung zum Geräte- und Produktsicherheitsgesetz (9. GPSGV)

Abbildung 5: Tabelle Europäische Richtlinien umgesetzt in deutsches Recht

5 Technische Anforderungen an Steuerungen

5.1 Aus Rechtsvorschriften

5.1.1 Anhang I Maschinenrichtlinie

In diesem Kapitel werden nur die Anforderungen an Steuerungen behandelt, die nicht von den bereits in Kapitel 4.2.3 angeführten Spezial-Richtlinien abgedeckt werden.

Die Maschinenrichtlinie behandelt die Anforderungen an Steuerungen in Anhang I Nr. 1.2.:

- 1.2.1. Sicherheit und Zuverlässigkeit von Steuerungen
- 1.2.2. Stellteile
- 1.2.3. Ingangsetzen
- 1.2.4. Stillsetzen
- 1.2.5. Betriebsartenwahlschalter
- 1.2.6. Störung der Energieversorgung
- 1.2.7. Störung des Steuerkreises
- 1.2.8. Software

Weitere Anforderungen finden sich in

- 1.4.2. Besondere Anforderungen an trennende Schutzeinrichtungen
- 1.4.3. Besondere Anforderungen an nichttrennende Schutzeinrichtungen
- 1.5.1 Gefahr durch elektrische Energie
- 3.3.3. Stillsetzen (mobile Arbeitsmittel)

Insbesondere aus den Anforderungen nach den Nummern 1.2.1 (siehe Kapitel 5.1.1.1) und 1.2.7 (siehe Kapitel 5.1.1.2) ergibt sich für den Maschinenhersteller bei der Verwendung der in dieser Arbeit behandelten Standard-SPS mit Diagnosefunktionen bzw. dem Sicherheitsbauteilhersteller, der eine solche SPS als Sicherheitsbauteil einzeln in Verkehr bringt, die zwingende Notwendigkeit, die sicherheitstechnische Funktion dieser Steuerung zu gewährleisten. Maßstab hierfür ist nach den Vorbemerkungen des Anhang I MRL der Stand der Technik. Die Verwendung der im Kapitel 10 beschriebenen Tests, soll den Hersteller in die Lage versetzen, diese rechtlichen Anforderungen ohne großen wirtschaftlichen Aufwand mit einer Standard-SPS zu erfüllen und soll insofern auch den Stand der Technik an dieser Stelle verbessern.

Die restlichen, o. a. steuerungsrelevanten Anforderungen beziehen sich auf die im Rahmen der Maschinenfunktion von der Steuerung auszuführenden Funktionen und haben insofern mit dem hier behandelten Thema nur indirekt zu tun.

5.1.1.1 Sicherheit und Zuverlässigkeit von Steuerungen

Anhang I der MRL fordert, dass Steuerungen sicher und zuverlässig funktionieren müssen:

1.2.1. Sicherheit und Zuverlässigkeit von Steuerungen

„Steuerungen sind so zu konzipieren und zu bauen, dass sie sicher und zuverlässig funktionieren und somit keine gefährlichen Situationen entstehen.
Insbesondere müssen sie so konzipiert und gebaut sein, dass

- *sie den zu erwartenden Betriebsbeanspruchungen und Fremdeinflüssen standhalten;*
- *Fehler in der Logik zu keiner gefährlichen Situation führen." [26]*

Für die Programmierung einer Standard-SPS mit Diagnosefunktionen bedeutet dies, dass Fehler im Programmablauf, die zu gefährlichen Situationen an der Maschine führen können, durch das Programm rechtzeitig erkannt werden müssen. Das Programm muss dann so reagieren, dass diese gefährlichen Situationen vermieden werden.

5.1.1.2 Störung des Steuerkreises

Anhang I der MRL fordert in Nr. 1.2.7, dass Störungen im Steuerkreis nicht zu gefährlichen Situationen führen:

„Ein Defekt in der Logik des Steuerkreises, eine Störung oder Beschädigung des Steuerkreises darf nicht zu gefährlichen Situationen führen.
Insbesondere ist folgendes auszuschließen:

- *unbeabsichtigtes Ingangsetzen;*
- *Nichtausführung eines bereits erteilten Befehls zum Stillsetzen;*
- *Herabfallen oder Herausschleudern eines beweglichen Maschinenteils oder eines von der Maschine gehaltenen Werkstücks;*
- *Verhinderung des automatischen oder manuellen Stillsetzens von beweglichen Teilen jeglicher Art;*
- *Ausfall von Schutzeinrichtungen." [26]*

Die in dieser Arbeit behandelte Standard-SPS ist dafür gedacht in einen Steuerkreis einer Maschine eingebaut zu werden. Defekte in der Logik dieses Steuerkreises dürfen nach den o. a. Anforderungen nicht zu gefährlichen Situationen führen. Das bedeutet:

- Alle Logik-Bauteile in der Steuerung müssen sicher funktionieren,

  oder

- im Steuerkreis wird sichergestellt, dass sicherheitsrelevante Fehler einzelner Bauteile erkannt und deren Auswirkung verhindert werden.

Für die Programmierung einer Standard-SPS mit Diagnosefunktionen findet der zweite Gedankenstrich Anwendung, (siehe Kapitel 10.8) da eine Eigensicherheit wie im ersten Gedankenstrich gefordert bei einer Standard-SPS nicht gegeben ist. Eine Eigensicherheit bietet zur Zeit nur eine Sicherheits-SPS.

5.1.2 Anhang III EMV Richtlinie

Die grundlegenden Anforderungen an SPSen sind:

- Begrenzung der elektromagnetischen Störaussendung auf ein solches Maß, dass keine anderen Betriebsmittel – insbesondere Funk- und Telekommunikationsendgeräte – unzulässig beeinflusst werden.
- Die SPS darf nicht selbst durch andere Betriebsmittel in ihrer Funktion unzulässig beeinflusst werden. (Mindeststörfestigkeit gegen äußere Störungen)

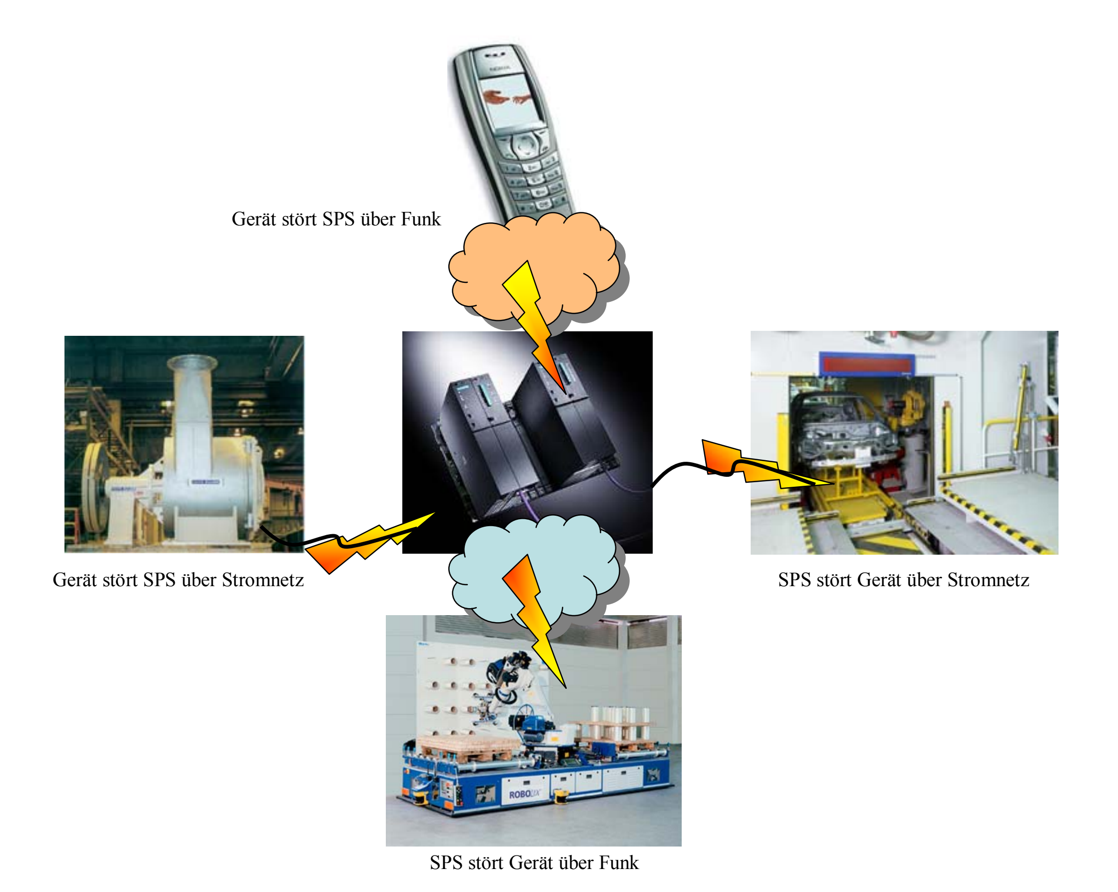


Abbildung 6: Störmöglichkeiten in Verbindung mit einer SPS

Dies wird ausführlich in Anhang III der EMV-Richtlinie [22] dargestellt:

> *„Erläuterndes Verzeichnis der wesentlichen Schutzanforderungen*
> *Der Höchstwert der von den Geräten ausgehenden elektromagnetischen Störungen muss so bemessen sein, dass der Betrieb insbesondere folgender Geräte nicht beeinträchtigt wird:*
> *a) private Ton- und Fernsehrundfunkempfänger,*
> *b) Industrieausrüstungen,*
> *c) mobile Funkgeräte,*
> *d) kommerzielle mobile Funk- und Funktelefongeräte,*
> *e) medizinische und wissenschaftliche Apparate und Geräte,*
> *f) informationstechnologische Geräte,*
> *g) Haushaltsgeräte und elektronische Haushaltsausrüstungen,*
> *h) Funkgeräte für die Luft- und Seeschiffahrt,*
> *i) elektronische Unterrichtsgeräte,*
> *j) Telekommunikationsnetze und -geräte,*
> *k) Sendegeräte für Ton-und Fernsehrundfunk,*
> *l) Leuchten und Leuchtstofflampen.*
> *Die – insbesondere unter den Buchstaben a) bis I) genannten – Geräte müssen so beschaffen sein, dass sie in einem normalen EMV-Umfeld ein angemessenes Störfestigkeitsniveau an ihrem Einsatzort aufweisen, damit sie unter Berücksichtigung der Werte hinsichtlich der Störung, die von den Geräten ausgeht, die den Normen des Artikels 7 entsprechen, ohne Beeinträchtigung betrieben werden können.*
> *Die für einen bestimmungsgemäßen Betrieb des Gerätes erforderlichen Angaben müssen in der beigefügten Bedienungsanleitung enthalten sein.“ [22]*

Nach Artikel 7 der EMV-Richtlinie [22] wird bei Einhaltung harmonisierter Normen (siehe Kapitel 4.2.2.2) von der Übereinstimmung mit den Bestimmungen der Richtlinie ausgegangen. Diese Wirkung

kann ggf. auch durch die Anwendung bestimmter nationaler Normen erreicht werden. In den anderen Fällen muss im Konformitätsbewertungsverfahren eine benannte Stelle eingeschaltet werden (ähnlich Anhang IV Sicherheitsbauteile).

5.2 Aus den Regeln der Technik

Im Kapitel 4.2.2.1 wurde die Wichtigkeit der Anwendung harmonisierter Normen mit Konformitätsvermutung dargelegt. Für die Betrachtung der Sicherheit einer SPS ist dies zur Zeit die DIN EN 954-1 [14], die Steuerungen nach deren Architektur in Verbindung mit den angewandten Fehlererkennungsmethoden in vorgegebene Steuerungskategorien einteilt. Diese wird in absehbarer Zeit durch die prEN ISO 13849-1 [15] abgelöst, welche zusätzlich die Wirksamkeit verwendeter Tests in die Bewertung einbezieht und die Ausfallraten der einzelnen Bauteile berücksichtigt.

In diesem Kapitel werden die Bewertungsverfahren dieser Normen erläutert. Zusätzlich wird die nicht nach MRL harmonisierte Norm DIN EN 61508 [17] vorgestellt, die zusammen mit der DIN EN 954-1 die Grundlage für die prEN ISO 13849-1 darstellt.

Als vierte Norm wird die ebenfalls nicht nach MRL harmonisierte Norm DIN EN 61131 [16] vorgestellt, die unter anderem die in dieser Diplomarbeit verwendete Programmiersprache und den Aufbau von SPSen die diese Programmiersprache verwenden können beschreibt. Durch die Anwendung dieser Norm ist sichergestellt, dass die geschriebenen Programmbausteine auf möglichst viele SPSen unterschiedlichster Hersteller portierbar sind.

5.2.1 Einteilung von Steuerungen in Steuerungskategorien (DIN EN 954-1)

Nach Kapitel 1 – „Anwendungsbereich" – dieser Norm ist sie auf Steuerungen und somit auf SPSen anwendbar:

> *„[…] Sie [die Norm] gilt für alle sicherheitsbezogenen Teile von Steuerungen, unabhängig von der verwendeten Energieart, z. B. elektrisch, hydraulisch, pneumatisch, mechanisch. […] Sie [die Norm] gilt für alle Maschinen im gewerblichen und nichtgewerblichen Bereich. […]" [14]*

Die Norm beschäftigt sich mit der Einteilung von Steuerungen in Steuerungskategorien und mit der Auswahl einer geeigneten Steuerungskategorie anhand eines Risikographens (siehe Abbildung 8).

Kategorien sind im Kapitel 3 Absatz 2 wie folgt definiert:

> *„3.2 Kategorie: Einteilung der sicherheitsbezogenen Teile einer Steuerung in Bezug auf ihre Widerstandsfähigkeit gegen Fehler und ihr Verhalten im Fehlerfall, die aufgrund der strukturellen Anordnung der Teile und/oder deren Zuverlässigkeit erreicht wird." [14]*

Es gibt fünf Kategorien. Die Norm gibt eine Kurzbeschreibung aller Kategorien in Kapitel 6 Absatz 1:

> *„6.1: […] Kategorie B ist die Basiskategorie. Das Auftreten eines Fehlers kann zum Verlust der Sicherheitsfunktion führen. In Kategorie 1 wird eine höhere Widerstandsfähigkeit gegen Fehler überwiegend durch Auswahl und Verwendung von Bauteilen erreicht. In den Kategorien 2, 3 und 4 wird eine verbesserte Leistungsfähigkeit hinsichtlich einer vorgegebenen Sicherheitsfunktion überwiegend durch Strukturverbesserungen an dem sicherheitsbezogenen Teil der Steuerung erreicht. Kategorie 2 sieht vor, dass die Ausführung der Sicherheitsfunktion in regelmäßigen Abständen geprüft wird. Kategorien 3 und 4 sehen zu diesem Zweck die Sicherstellung vor, dass das Auftreten eines einzelnen Fehlers nicht zum Verlust der Sicherheitsfunktion führt. […]" [14]*

Eine gute Übersicht über die einzelnen Abstufungen zwischen den Kategorien ist aus der Tabelle 2 der Norm (Abbildung 7) ersichtlich, in der die vollständigen Anforderungen der Kategorien zusammengefasst sind:

Kategorie	Kurzfassung der Anforderungen	Systemverhalten	Prinzipien zum Erreichen der Sicherheit
B	Die sicherheitsbezogenen Teile von Steuerungen und/oder ihre Schutzeinrichtungen, als auch ihre Bauteile müssen in Übereinstimmung mit den zutreffenden Normen so gestaltet, gebaut, ausgewählt, zusammengestellt und kombiniert werden, dass sie den zu erwartenden Einflüssen standhalten können.	Das Auftreten eines Fehlers kann zum Verlust der Sicherheitsfunktion führen.	**überwiegend durch Auswahl von Bauteilen charakterisiert**
1	Die Anforderungen von B müssen erfüllt sein. Bewährte Bauteile und bewährte Sicherheitsprinzipien müssen angewendet werden.	Das Auftreten eines Fehlers kann zum Verlust der Sicherheitsfunktion führen, aber die Wahrscheinlichkeit des Auftretens ist geringer als in Kategorie B.	
2	Die Anforderungen von B und die Verwendung bewährter Sicherheitsprinzipien müssen erfüllt sein. Die Sicherheitsfunktion muss in geeigneten Zeitabständen durch die Maschinensteuerung geprüft werden.	- Das Auftreten eines Fehlers kann zum Verlust der Sicherheitsfunktion zwischen den Prüfabständen führen. - Der Verlust der Sicherheitsfunktion wird durch die Prüfung erkannt.	**überwiegend durch die Struktur charakterisiert**
3	Die Anforderungen von B und die Verwendung bewährter Sicherheitsprinzipien müssen erfüllt sein. Sicherheitsbezogene Teile müssen so gestaltet sein, dass: - ein einzelner Fehler in jedem dieser Teile nicht zum Verlust der Sicherheitsfunktion führt, und - wann immer in angemessener Weise durchführbar, der einzelne Fehler erkannt wird.	- Wenn der einzelne Fehler auftritt, bleibt die Sicherheitsfunktion immer erhalten. - Einige aber nicht alle Fehler werden erkannt. - Eine Anhäufung unerkannter Fehler kann zum Verlust der Sicherheitsfunktion führen.	**überwiegend durch die Struktur charakterisiert**

Kategorie	Kurzfassung der Anforderungen	Systemverhalten	Prinzipien zum Erreichen der Sicherheit
4	Die Anforderungen von B und die Verwendung bewährter Sicherheitsprinzipien müssen erfüllt sein. Sicherheitsbezogene Teile müssen so gestaltet sein, dass: - ein einzelner Fehler in jedem dieser Teile nicht zum Verlust der Sicherheitsfunktion führt, und - der einzelne Fehler bei oder vor der nächsten Anforderung an die Sicherheitsfunktion erkannt wird, oder, wenn dies nicht möglich ist, darf eine Anhäufung von Fehlern dann nicht zum Verlust der Sicherheitsfunktion führen.	- Wenn Fehler auftreten, bleibt die Sicherheitsfunktion immer erhalten. - Die Fehler werden rechtzeitig erkannt, um einen Verlust der Sicherheitsfunktion zu verhindern.	**überwiegend durch die Struktur charakterisiert**

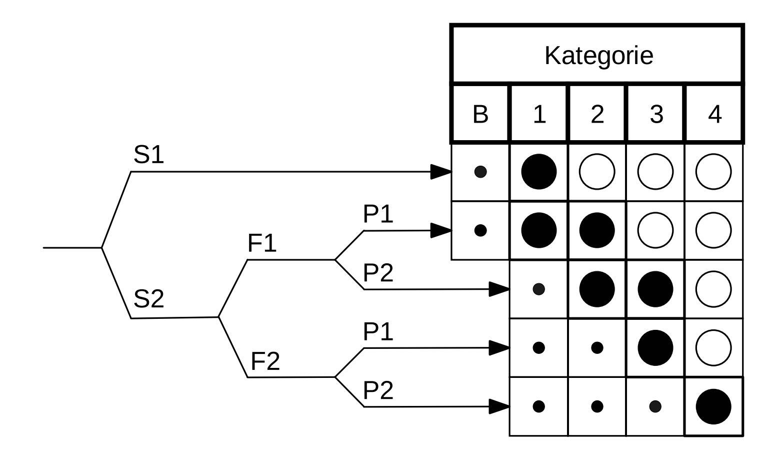
Abbildung 7: DIN EN 954-1 Tabelle 2: Kurzfassung der Anforderungen für Kategorien [14]

Aus dieser Tabelle und der Kurzbeschreibung wird deutlich, dass die Norm die Sicherheit von Steuerungen anhand der Architektur der Steuerung in Verbindung mit den angewandten Fehlererkennungsmethoden beurteilt. Die Güte dieser Fehlererkennungsmethoden, oder die Ausfallrate der verwendeten Bauteile fließt allerdings nicht in die Bewertung mit ein.

Diese Einteilung sicherheitsgerichteter Teile der Steuerung in einzelne Kategorien ist der Hauptteil der Norm. Zusätzlich befindet sich im Anhang B „Hinweise zur Auswahl von Kategorien“ ein Risikograph (Abbildung 8), der dem Konstrukteur hilft, die notwendige Steuerungskategorie für seine Maschine auszuwählen. Abweichungen von den vorgeschlagenen Kategorien sind möglich, müssen aber begründet werden. In der Einleitung des Anhangs B findet sich der Satz:

„Die Hinweise, die in diesem Anhang gegeben werden, sollten als Teil der in EN 1050 beschriebenen Risikobeurteilung betrachtet werden und nicht als Ersatz dafür.“ [14]

Dieser Graph in Abbildung 8 ist also nur als Hilfe für die Auswahl der richtigen Steuerungskategorie gedacht, nicht als Ersatz einer kompletten Risikobeurteilung einer Maschine oder Anlage.



● Bevorzugte Kategorien für Bezugspunkte

• Mögliche Kategorien die zusätzliche Maßnahmen erfordern

○ Maßnahmen, die in Bezug auf das zutreffende Risiko überdimensioniert sein können

S Schwere der Verletzung

S1 Leichte (üblicherweise reversieble) Verletzung

S2 Schwere (üblicherweise irreversieble) Verletzung, einschließlich Tot

F Häufigkeit und / oder Dauer der Gefährdungsexposition

F1 Selten bis öfter und / oder kurze Dauer der Exposition

F2 Häufig bis dauernd und / oder lange Dauer der Exposition

P Möglichkeit zur Vermeidung der Gefährdung

Möglich unter bestimmten Bedingungen

Kaum möglich

Abbildung 8: Risikograph nach DIN EN 954-1 [14]

Zur Auswahl der im konkreten Einzelfall notwendigen Steuerungskategorie der Sicherheitssteuerung wird zuerst die mögliche Schwere einer Verletzung festgestellt, die durch die Maschine ohne die zu steuernde Schutzeinrichtungen entstehen kann. Es wird bei möglichen Verletzungen nur zwischen leichten, üblicherweise reversiblen (S1) und schweren, üblicherweise irreversiblen Verletzungen, einschließlich Tod (S2), unterschieden.

In der zweiten Stufe wird die Dauer berücksichtigt, die der Benutzer während des zu betrachtenden Arbeitsvorganges der Gefahr ausgesetzt ist. Auch hier wird nur zwischen selten bis öfter (F1) und häufig bis dauernd (F2) unterschieden. Zu beachten ist hierbei, dass eine dauernde Gefährdung während eines seltenen Betriebszustands, z.B. der Wartung, trotzdem noch eine dauernde Gefährdung im Sinne dieser Norm ist. Die Norm gibt hierzu ein Beispiel an.

In der letzten Stufe des Risikographen wird die Möglichkeit der Vermeidung der Gefährdung berücksichtigt. Entweder ist der Benutzer oder eine dritte Person in der Lage, die drohende Gefahr zu erkennen und abzuwenden, bzw. der Benutzer kann sich in Sicherheit bringen (P1), oder die Abwendung der Gefahr ist kaum möglich (P2).

Diese Abschätzungen führen zur Auswahl der notwendigen Steuerungskategorie.

Eine Steuerung der Kategorie B der auch (nach Kapitel 7.1.1) die Standard-SPS angehört, kann auf Basis des Risikographen nur, unter Zuhilfenahme zusätzlicher Maßnahmen, für Maschinen eingesetzt werden.

5.2.2 Bewerten der Wirksamkeit von Diagnosefunktionen (DIN EN 61508)

Aus dem Vorwort der Norm:

> *„Diese Internationale Norm behandelt diejenigen Gesichtspunkte, die zu betrachten sind, wenn elektrische / elektronische / programmierbar elektronische Systeme (E/E/PES) zur Ausführung von Sicherheitsfunktionen eingesetzt werden.“ [17]*

Diese Norm, die als eine der Grundlagen der in Kapitel 5.2.3 beschriebenen prEN ISO 13849-1 dient, ist für Steuerungen bzw. Rechnersysteme gedacht, die Sicherheitsfunktionen erfüllen. Sie gibt dabei u. a. in Tabellen Richtwerte an, welche Art von Tests (Diagnosefunktionen) im Rechner selbst zu welcher Fehleraufdeckungsrate führen.

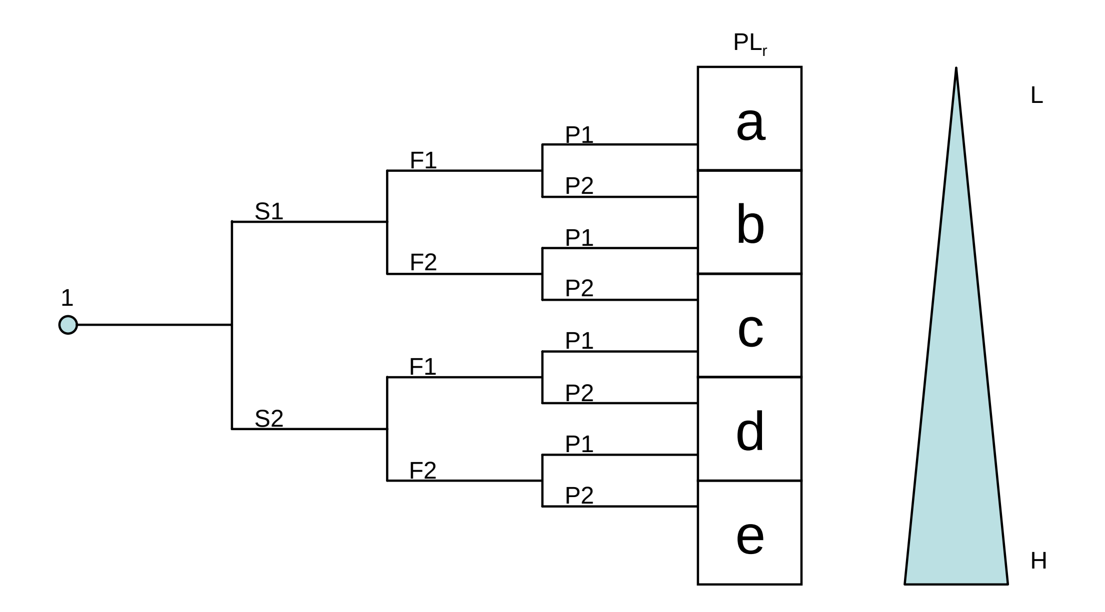
Die Vorgehensweise der Norm, zur Ermittlung der Sicherheit einer Steuerung, wurde in dieser Diplomarbeit nicht verwendet. Da die Norm aber eine Auflistung von Tests und deren Wirksamkeit enthält, konnte sie genutzt werden, um die Wirksamkeit einiger erarbeiteter Tests festzustellen, indem diese mit den in dieser Norm beschriebenen Tests verglichen wurden.

5.2.3 Einteilung von Steuerungen in Performance Level (prEN ISO 13849-1)

Wie im Kapitel 5.2.1 beschrieben, setzt die heutige Bewertung der Sicherheit einer Steuerung allein auf der Architektur der Steuerung in Verbindung mit den angewandten Fehlererkennungsmethoden auf. Um die verschiedenen Wirkungsgrade von Tests und die Ausfallsicherheit von Bauteilen mit zu berücksichtigen wird zur Zeit auf Ebene der „International Organization for Standardization“ (ISO) in Zusammenarbeit mit dem Europäischen Komitee für Normung (Comité Européen de Normalisation – CEN) an der prEN ISO 13849-1 gearbeitet. Diese wird die Inhalte der DIN EN 954-1 (Kapitel 5.2.1) mit dem Ansatz der Berücksichtigung der Wirksamkeit verwendeter Tests und der Ausfallraten der einzelnen Bauteile aus der nicht harmonisierten DIN EN 61508 (Kapitel 5.2.2) vereinigen.

Die prEN ISO 13849-1 soll in Kürze die Norm DIN EN 954-1 als nach Maschinenrichtlinie harmonisierte Norm ersetzen. Nach ihrer Verabschiedung wird es eine Übergangsfrist von drei Jahren geben. Da sie noch nicht verabschiedet wurde existiert noch keine offizielle Übersetzung. Um die Lesbarkeit dieser Arbeit zu erhöhen wurden die offiziellen englischen Abkürzungen beibehalten, deren Langtext aber ins Deutsche übersetzt. Als Quelle für diese Übersetzungen wurde die Zusammenfassung des Vortrages von Herrn Dipl.-Ing. Thomas Bömer und Herrn Dipl.-Ing. Karl-Heinz Büllesbach [2] verwendet.

Der Risikograph der Norm prEN ISO 13849-1 führt nicht mehr zu Vorschlägen für Kategorien, sondern verlangt nun die Erfüllung eines „benötigten Performance Level“ (Performance Level required – PL_r).



S Schwere der Verletzung

S1 Leichte (üblicherweise reversieble) Verletzung

S2 Schwere (üblicherweise irreversieble) Verletzung, einschließlich Tot

F Häufigkeit und / oder Dauer der Gefährdungsexposition

F1 Selten bis öfter und / oder kurze Dauer der Exposition

F2 Häufig bis dauernd und / oder lange Dauer der Exposition

P Möglichkeit zur Vermeidung der Gefährdung

Möglich unter bestimmten Bedingungen

Kaum möglich

Abbildung 9: Risikograph nach prEN ISO 13849-1 [15]

Die einzelnen Abschätzungen S, F und P, des von der betrachteten Gefahrstelle der Maschine ausgehenden Risikos sind identisch mit denen, die bereits im Risikographen der DIN EN 954-1 in Abbildung 8 benutzt wurden. Der Risikograph der prEN ISO 13849-1 (Abbildung 9) unterscheidet im Vergleich mit dem Risikographen der DIN EN 954-1 (Abbildung 8) stärker bei leichten Verletzungen und überlässt die Wahl des PL_r (vorher Kategorie) nicht mehr dem Hersteller.

Je höher der tatsächlich vorhandene Performance Level (PL) ist, desto geringer ist die durchschnittliche Wahrscheinlichkeit des Auftretens eines gefährlichen Ausfalls (siehe Abbildung 10). Ein gefährlicher Ausfall bezeichnet einen unerkannt auftretenden Fehler, der zum Versagen einer Sicherheitsfunktion führt.

Performance Level (PL)	Durchschnittliche Wahrscheinlichkeit des Auftretens eines gefährlichen Ausfalls [1/h]
a	$\geq 10^{-5}$ bis $> 10^{-4}$
b	$\geq 3 * 10^{-6}$ bis $> 10^{-5}$
c	$\geq 10^{-6}$ bis $> 3 * 10^{-6}$
d	$\geq 10^{-7}$ bis $> 10^{-6}$
e	$\geq 10^{-8}$ bis $> 10^{-7}$

Abbildung 10: prEN ISO 13849-1 Tabelle 3: Performance Level [15]

Die prEN ISO 13849-1 beginnt beim Bewerten einer Steuerung wie die DIN EN 954-1 mit der Einteilung der Steuerung in Kategorien.

Dabei behält sie die Einteilung der Steuerung in Steuerungskategorien aus der DIN EN 954-1 bei. Lediglich die Kategorie 2 erhält eine höhere Anforderung, da hier eine zusätzliche Testeinrichtung mit einem separaten Abschaltpfad gefordert ist. Zusätzlich wird in der prEN ISO 13849-1 ab der Steuerungskategorie 2 eine Bewertung der Ausfälle gemeinsamer Ursachen (Common Cause Failure – CCF) gefordert. Diese Bewertung wird am Ende dieses Kapitels erläutert.

Auf der Basis der ermittelten Steuerungskategorie wird später der Performance Level ermittelt. Dazu muss auch noch die mittlere Zeit bis zu einem gefährlichen Ausfall des Geräts (Mean Time To Failure dangerous – $MTTF_d$) und die Fehleraufdeckungsrate im Durchschnitt (Diagnostic Coverage average – DC_{avg}) bestimmt werden. Der Gesamtzusammenhang ist in Abbildung 11 dargestellt.

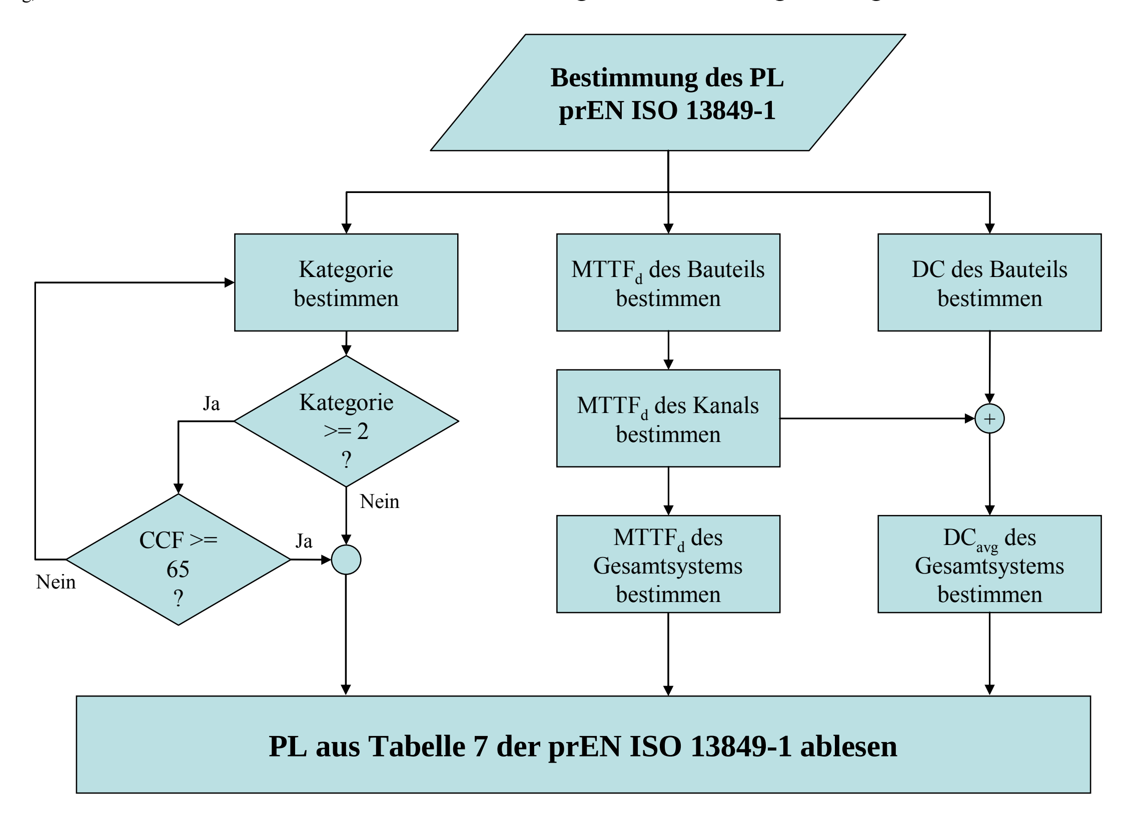


Abbildung 11: Bestimmung des Performance Levels nach prEN ISO 13849-1

Für die Bestimmung des $MTTF_d$ stehen verschiedene Methoden zur Auswahl, die in Abbildung 12 zusammengefasst und im folgenden Text erläutert sind.

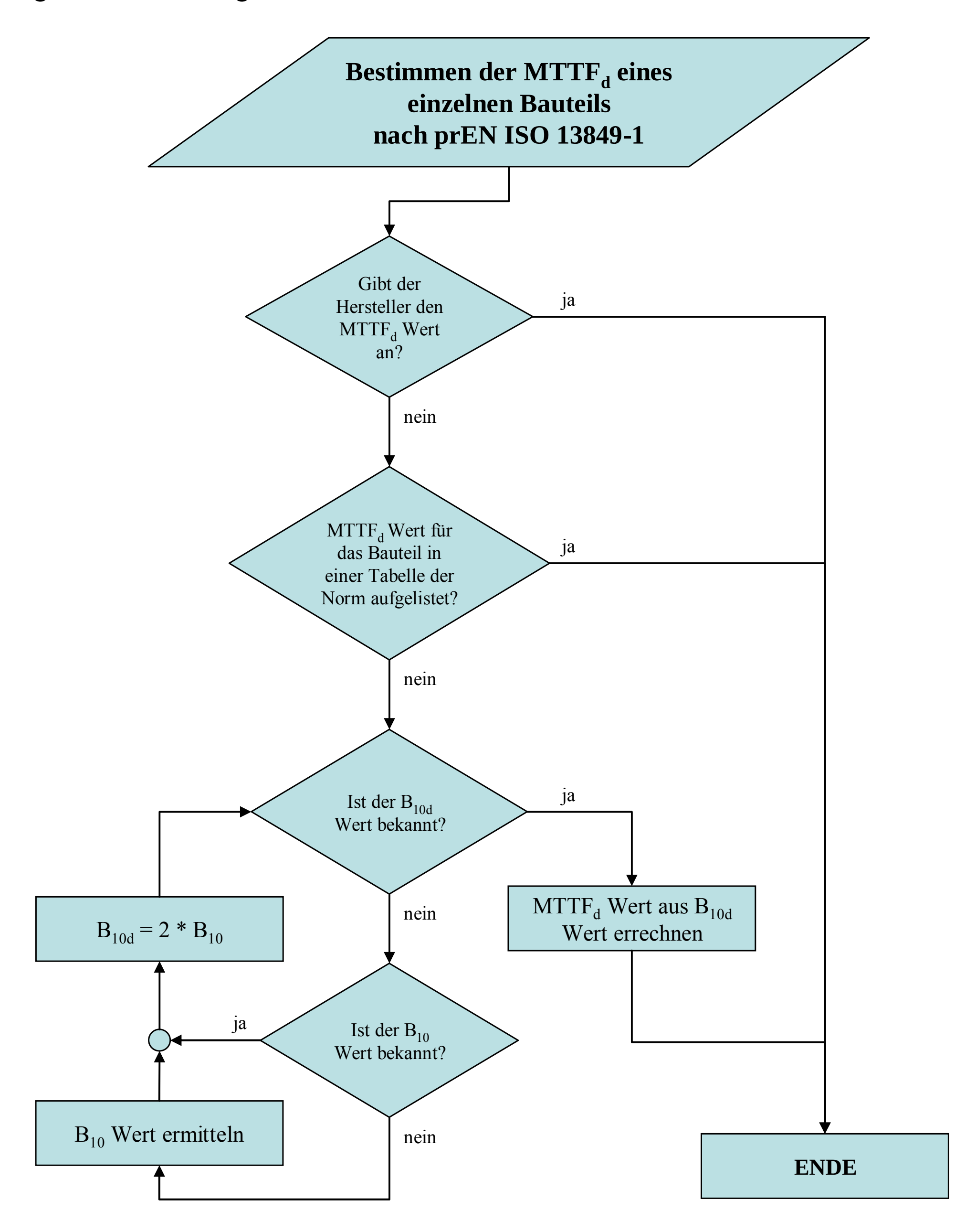


Abbildung 12: Darstellung der Bestimmung des $MTTF_d$ nach prEN ISO 13849-1

Es ist vorgesehen, dass zukünftig die Werte der einzelnen Bauteile für MTTF und $MTTF_d$ von den Herstellern mitgeliefert werden. Bis dahin führt die prEN ISO 13849-1 im Anhang C und D Werte in Tabellen auf, die als Näherungswerte für heute benutzte Standardbauteile benutzt werden können. Für pneumatische sowie elektromechanische Bauteile, z. B. in der SPS eingesetzte Relais, kann als Grundlage für die Berechnung des $MTTF_d$ Werts der B_{10d} Wert, das heißt die Anzahl der Schaltspiele, bei der 10% der Stichproben mit einem gefährlichen Fehler ausfallen, benutzt werden. (siehe Formel aus Abbildung 13) Ist der B_{10d} Wert nicht bekannt, kann angenommen werden das 50% aller Ausfälle gefährlich sind und damit gilt $B_{10d} = 2*B_{10}$. B_{10} ist die Anzahl der Schaltspiele, bei der 10% der

Stichproben ausfallen. Zur Bestimmung des B_{10} Wertes müssen ggf. entsprechende Versuche durchgeführt werden.

Zur Berechnung des $MTTF_d$ Wertes aus dem B_{10d} Wert, muss aus den Werten für

- d_{op} (days of operation per year), Anzahl an Tagen pro Jahr, an denen das Gerät läuft
- h_{op} (hours of operation per day), Stunden pro Tag an denen das Gerät läuft
- t_{cycle} (time between the beginning of two successive cycles), Zeitdauer eines Schaltspiel-Zyklus

der Wert

- n_{op} (numbers of operation per year), Anzahl der Schaltspiel-Zyklen pro Jahr

errechnet werden.

$$MTTF_d = \frac{B_{10d}}{0{,}1 * n_{op}}$$

$$n_{op} = \frac{d_{op} * h_{op} * 3.600 \frac{s}{h}}{t_{cycle}}$$

Abbildung 13: Formel zur Berechnung von $MTTF_d$ aus B_{10d} aus prEN ISO 13849-1 Anhang C.4.2 [15]

Um für Bauteile, die in Reihe miteinander verschaltet sind, einen gemeinsamen $MTTF_d$ zu berechnen, werden die Komplemente der einzelnen $MTTF_d$ addiert und hieraus wiederum das Komplement gebildet. (siehe Formel in Abbildung 14)

$$\frac{1}{MTTF_{d,gesamt}} = \sum_{i=1}^{N} \frac{1}{MTTF_{d,i}}$$

Abbildung 14: Formel zur Berechnung des gesamt MTTFd in Reihe geschalteter Bauteile aus prEN ISO 13849-1 Anhang D.1 [15]

Da die Norm in anderen Abschnitten davon ausgeht, dass alle Zeiten bis zum gefährlichen Ausfall gleich sind, findet sich im Anhang D der prEN ISO 13849-1 eine Formel, mit der ein mittlerer $MTTF_d$ für parallele Kanäle berechnet werden kann. (siehe Abbildung 15) C1 und C2 stehen für Kanal 1 bzw. 2 (zur Mehrkanaligkeit siehe nachfolgenden Absatz):

$$MTTF_{d,gesamt} = \frac{2}{3}\left[MTTF_{d,C1} + MTTF_{d,C2} - \frac{1}{\frac{1}{MTTF_{d,C1}} + \frac{1}{MTTF_{d,C2}}} \right]$$

Abbildung 15: Formel zur Berechnung des gesamt MTTFd paralleler Kanäle aus prEN ISO 13849-1 Anhang D.2 [15]

Die Abbildung 16 zeigt ein Beispiel für den Aufbau eines mehrkanaligen Steuerungssystems. In diesem Fall ist eine zweikanalige Steuerung nach Kategorie 4 dargestellt. Beide Kanäle arbeiten autark und sind nur zur Überwachung der Funktion des jeweils anderen Kanals miteinander verbunden. Durch einen solchen Aufbau ist gewährleistet, dass ein Fehler im Gesamtsystem sicher erkannt wird und keinen gefährlichen Zustand einleiten kann.

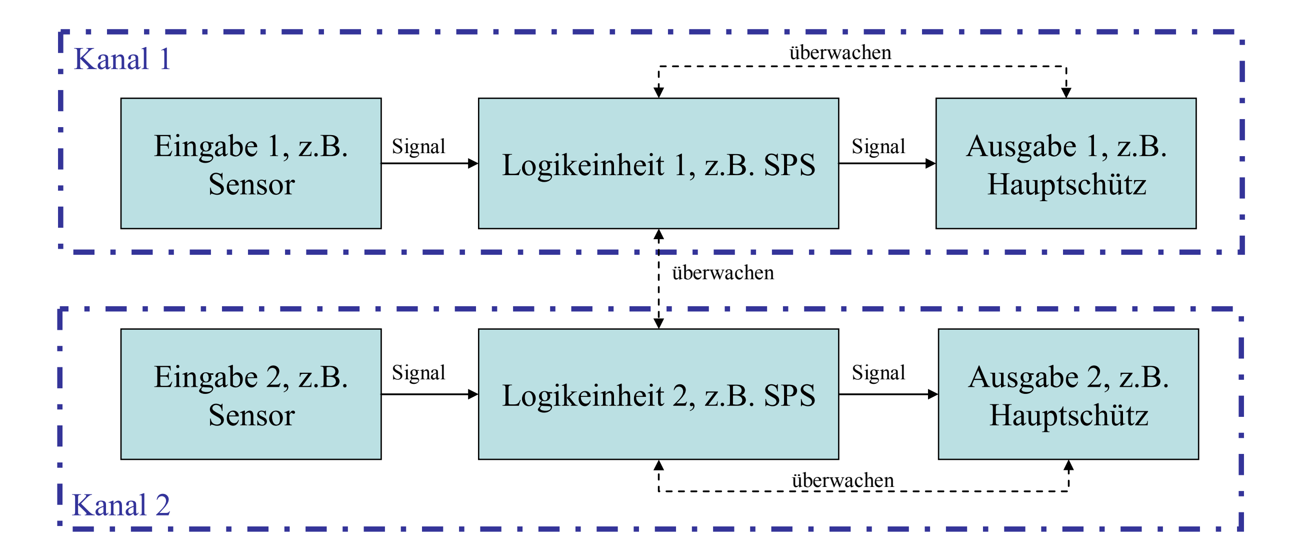


Abbildung 16: 2 Kanäle einer Kategorie 4 Schaltung

Die erreichten $MTTF_d$ werden dann, wie in der Tabelle in Abbildung 17 dargestellt, in eine von drei Stufen (niedrig, mittel, hoch) eingeteilt. Ein Bauteil oder Kanal eines Systems mit einer $MTTF_d$ von weniger als drei Jahre ist nicht zulässig. Eine $MTTF_d$ von mehr als 100 Jahren ist für ein Bauteil zulässig, wird für einen ganzen Kanal aber als nicht praktikabel erachtet und im weiteren Verlauf auf 100 Jahre begrenzt.

Bedeutung des $MTTF_d$ jedes Kanals	Bereich des $MTTF_d$ jedes Kanals
Niedrig	3 Jahre $\leq MTTF_d <$ 10 Jahre
Mittel	10 Jahre $\leq MTTF_d <$ 30 Jahre
Hoch	30 Jahre $\leq MTTF_d <$ 100 Jahre

Abbildung 17: prEN ISO 13849-1 Tabelle 5: Mittlere Zeit bis zum gefährlichen Ausfall ($MTTF_d$)

Im nächsten Schritt wird die Diagnosedeckungsgrad (DC – Diagnostic Coverage) bestimmt. Dabei handelt es sich, wie in Abbildung 18 dargestellt, um das Verhältnis der erkannten gefährlichen Fehler zu den unerkannten gefährlichen Fehlern.

$$DC = \frac{Fehler_{gefährlich,erkannt}}{Fehler_{gefährlich,gesamt}}$$

Abbildung 18: Begriffserläuterung DC

Aus den einzelnen DCs der getesteten Teile und deren $MTTF_d$ wird ein durchschnittlicher Diagnosedeckungsgrad DC_{avg} errechnet: (siehe Abbildung 19)

$$DC_{avg} = \frac{\sum_{i=1}^{N} \frac{DC_i}{MTTF_{d,i}}}{\sum_{i=1}^{N} \frac{1}{MTTF_{d,i}}} = MTTF_{d,gesamt} * \sum_{i=1}^{N} \frac{DC_i}{MTTF_{d,i}}$$

Abbildung 19: Formel zur Berechnung des durchschnittlichen DCs, nach prEN ISO 13849-1 Anhang E.2 [15], $MTTF_{d,\,gesamt}$ aus Abbildung 14

Dieser durchschnittliche DC wird wiederum in vier Klassen, von „keine Fehleraufdeckung“ bis „hohe Fehleraufdeckung“, eingeteilt, wie die Tabelle in Abbildung 20 zeigt:

Fehleraufdeckung	Wertebereich DC
keine	DC < 60%
niedrige	60% ≤ DC < 90%
mittlere	90% ≤ DC < 99%
hohe	99% ≤ DC

Abbildung 20: Bestimmung des Grads der Fehleraufdeckung nach prEN ISO 13849-1 [15]

Das erreichte Performance Level „a“ bis „e“ wird aus der Tabelle in Abbildung 21 bestimmt. Die Steuerungskategorie der Steuerung und der erreichte DC_{avg} der Tests ergeben die Spalte, der $MTTF_d$ der Kanäle ergibt die Zeile.

Kategorie	**B**	**1**	**2**	**2**	**3**	**3**	**4**
DC_{avg}	**keine**	**keine**	**niedrig**	**mittel**	**niedrig**	**mittel**	**hoch**
$MTTF_d$ jedes Kanals: niedrig	a	Nicht abgedeckt	a	b	b	c	Nicht abgedeckt
$MTTF_d$ jedes Kanals: mittel	b	Nicht abgedeckt	b	c	c	d	Nicht abgedeckt
$MTTF_d$ jedes Kanals: hoch	Nicht abgedeckt	c	c	d	d	d	e

Abbildung 21: prEN ISO 13849-1 Tabelle 7: Vereinfachte Bestimmung des Performance Levels [15]

Für konkrete Beispiele zur Anwendung der Tabelle siehe Abbildung 34 aus Kapitel 7 sowie Abbildung 75 aus Kapitel 12.2.

Ab einer „Kategorie 2 Steuerung“ sieht die prEN ISO 13849-1 zusätzlich eine Prüfung der CCF nach der Tabelle in Abbildung 22 vor. Dieser Wert geht jedoch nicht in die Berechnung des PL ein. Stattdessen muss die Addition von Maßnahmen, bzw. der entsprechenden Werte mindestens einen Schwellenwert von 65 Punkten erreichen. Ansonsten ist diese Steuerung nicht als Kategorie 2 oder höher einzuordnen.

Maßnahme	Punkte
Trennung der Signalpfade	15
Diversität	20
Schutz gegen z.B. Überspannung / Überdruck	15
Bewährte Bauteile	5
FMEA	5
Kompetenz / Training der Entwickler	5
EMV oder Filterung des Druckmediums und Schutz gegen Verschmutzung	25
Temperatur, Feuchte, Schock, Vibration usw.	10

Abbildung 22: Auflistung von Schutzmöglichkeiten gegen CCF nach prEN ISO 13849-1 Tabelle F.1, entnommen aus [2] Abbildung 31

5.2.4 Speicherprogrammierbare Steuerungen (DIN EN 61131)

Die DIN EN 61131 fasst Anforderungen für moderne SPS-Systeme zusammen. Das Ziel der Norm ist es, diese Anforderungen für alle auf dem Markt befindlichen SPSen zu vereinheitlichen. Wie in der Einführung in Kapitel 5.2 beschrieben enthält diese Norm die in dieser Diplomarbeit verwendete Programmiersprache (in ihrem Teil drei) und beschreibt weiterhin den generellen Aufbau von SPSen (in ihrem Teil zwei). In der Norm sind gängige Konzepte älterer SPS-Programmiersprachen und Erweiterungen um moderne Programmiersprachen enthalten. Die einzelnen Sprachen werden in Kapitel 8 dieser Diplomarbeit beschrieben. Hier wird auch dargelegt, warum die Sprache „Anweisungsliste“ (AWL) für die Tests dieser Diplomarbeit ausgewählt wurde.

Aus dem Teil zwei der Norm wurden besonders die Pflichten des Herstellers zur Umsetzung von Selbsttests in der SPS betrachtet, da die Effizienz dieser Selbsttests bei der Bestimmung des Performance Levels nach prEN ISO 13849-1 berücksichtigt werden kann. Nach Kapitel 3 „Elektrische Anforderungen“, Abschnitt 3.11 Anforderungen an „Selbstprüfungen und Diagnosemöglichkeiten“ muss der Hersteller einer SPS grundsätzlich folgende Tests implementieren:

- *„eine Zeitüberwachung (Watchdog)*
- *eine Möglichkeit der Speicherprüfung (Hard- oder Software)*
- *eine Möglichkeit die Übertragung der Daten zwischen Speicher, Verarbeitungseinheit und den Ein-/Ausgabemodulen auf Richtigkeit zu überwachen*
- *eine Möglichkeit der Spannungsüberwachung (z. B. eine Sicherung)*
- *eine Zustandsüberwachung der Hauptverarbeitungseinheit*

Bei dem Auftauchen eines Fehlers in einem dieser Tests muss ein Alarm-Ausgang gesetzt werden.“ [16]

In dieser Diplomarbeit wird der Watchdog (Kapitel 10.3.1) nach Spiegelstrich eins, sowie die CRC-Prüfung (Kapitel 10.5.1) nach Spiegelstrich zwei und drei mit in die Tests einbezogen, da bei diesen ein Hersteller übergreifend einheitliches Verhalten angenommen werden kann. Andere vom Hersteller implementierte Tests können im konkreten Einzelfall zur weiteren Verbesserung des DC beitragen.

6 Funktionsbeschreibung einer SPS

Bei einer Speicherprogrammierbaren Steuerung (SPS) handelt es sich um einen Computer mit Ein- und Ausgabeschnittstellen, die für den Anschluss von Sensoren und Aktuatoren gedacht sind. Die SPS wird hauptsächlich in der Automatisierungstechnik zum Steuern und Regeln von Maschinen und Anlagen eingesetzt. Auf dem Markt existieren verschiedene SPS-Arten, die in Kapitel 6.1 beschrieben werden.

Im Rahmen dieser Diplomarbeit wurden die Selbsttests, die in Kapitel 10 beschrieben werden, auf der Soft-SPS (siehe Kapitel 6.1.2) von CoDeSys getestet. Die Ergebnisse wurden allerdings im Hinblick auf eine Verwendung in einer Schrank-SPS beurteilt, weshalb sich das Kapitel 6.2 mit dem generellen Aufbau einer Schrank-SPS befasst, ohne konkrete Gerätedaten anzugeben.

6.1 SPS Arten

Zur Zeit gibt es auf dem Markt drei Arten von SPS-Systemen. Die Schrank-SPS, die Soft-SPS, und die Slot-SPS.

Zusätzlich wird in Standard-SPS und Sicherheits-SPS (SSPS) unterschieden.

6.1.1 Schrank-SPS

Die Schrank-SPS ist die klassische SPS in der Automatisierungstechnik. Ihr Name kommt daher, dass sie, wie in Abbildung 23 zu sehen, im Schaltschrank einer Maschine installiert ist.

Die in dieser Diplomarbeit entwickelten Selbsttests sind im Hinblick auf den Hardwareaufbau einer solchen SPS programmiert. Ihr Aufbau wird im Kapitel 6.2 konkret beschrieben.

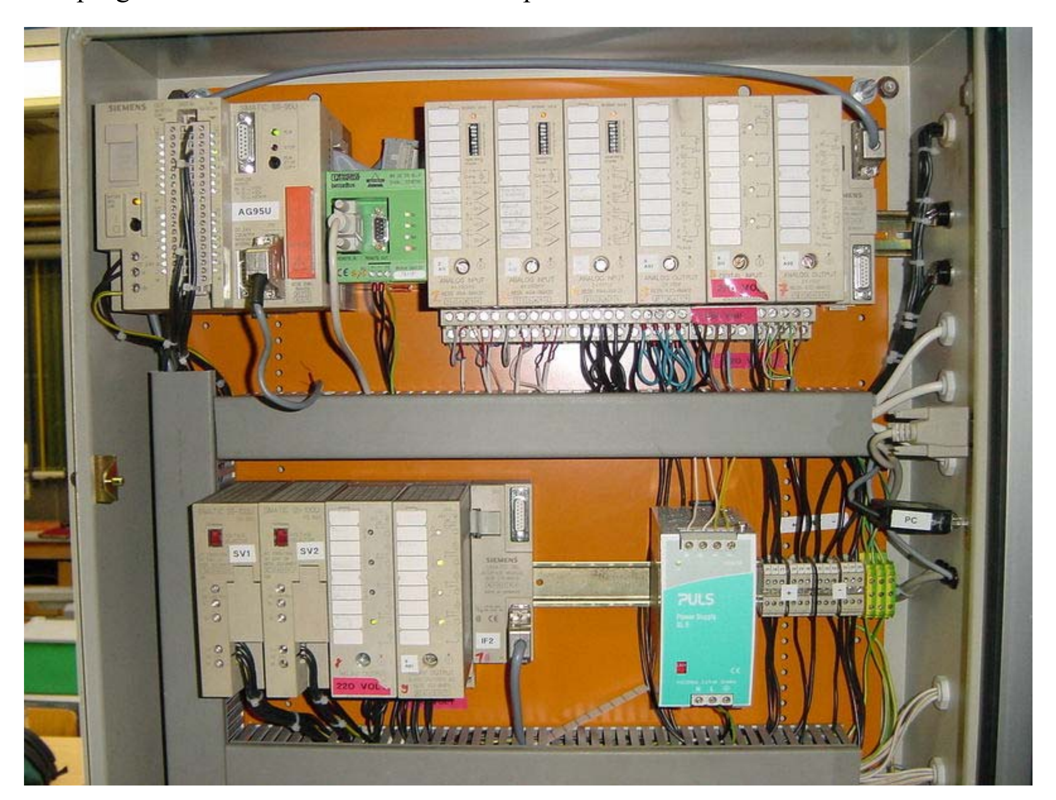


Abbildung 23: Schrank-SPS von SIEMENS

6.1.2 Soft-SPS

Eine Soft-SPS ist eine SPS, die auf einem PC simuliert wird. Dies bietet die Möglichkeit einer einfachen Visualisierung und einer schnellen Änderbarkeit der Programme bzw. wie in Abbildung 24 zu erkennen, der einfachen Portierbarkeit auf PC-ähnliche Geräte.

Die schnelle Änderbarkeit ist dadurch gegeben, dass SPS und Programmierumgebung in einem Gerät vereinigt sind. Die Portierbarkeit hängt vom Verbreitungsgrad der Soft-SPS-Umgebung auf anderen Geräten als dem PC, bzw. anderen Betriebssystemen als Windows ab.



Abbildung 24: Einfache Portierung eines Soft-SPS Programms von Windows auf einen PocketPC

Der größte Nachteil dieser SPS-Art liegt darin, dass ein PC nicht Echtzeitfähig ist. Das Betriebssystem und andere laufende Programme können die Simulation ausbremsen oder ganz anhalten. Aus diesem Grund allein ist eine sicherheitstechnische Ertüchtigung einer Soft-SPS nicht sinnvoll und wurde im Rahmen dieser Diplomarbeit nicht verfolgt. Für weitere Angaben zur Echtzeitfähigkeit siehe Kapitel 6.5.

Die in dieser Diplomarbeit entwickelten Selbsttests wurden zur Simulation zwar auf einer Soft-SPS getestet, sind aber nicht für den Betrieb auf einer Soft-SPS geeignet. Der Prozessortest ist auf den Befehlsumfang einer Schrank-SPS ausgelegt, der Prozessor eines Computers, auf dem die Soft-SPS läuft hat aber einen wesentlich größeren Befehlsumfang der ebenfalls zu gefährlichen Ausfällen führen kann. Auch der Speicher des Computers ist größer als der Teil, der aus der Soft-SPS heraus angesprochen werden kann. Bei einer Soft-SPS ist es aus diesem Grund empfehlenswert den gesamten Computer zu überwachen. Dazu ist ein Testprogramm, das speziell auf das Betriebssystem und die Hardware des Computers zugeschnitten ist, besser geeignet, als das in dieser Diplomarbeit entwickelte.

6.1.3 Slot-SPS

Die Slot-SPS ist eine Kombination der beiden vorgenannten Arten. Sie besteht aus der gleichen Hardware wie eine Schrank-SPS, ist aber nicht mit einem eigenen Gehäuse versehen, sondern auf einer Einsteckkarte für den PC aufgebaut.

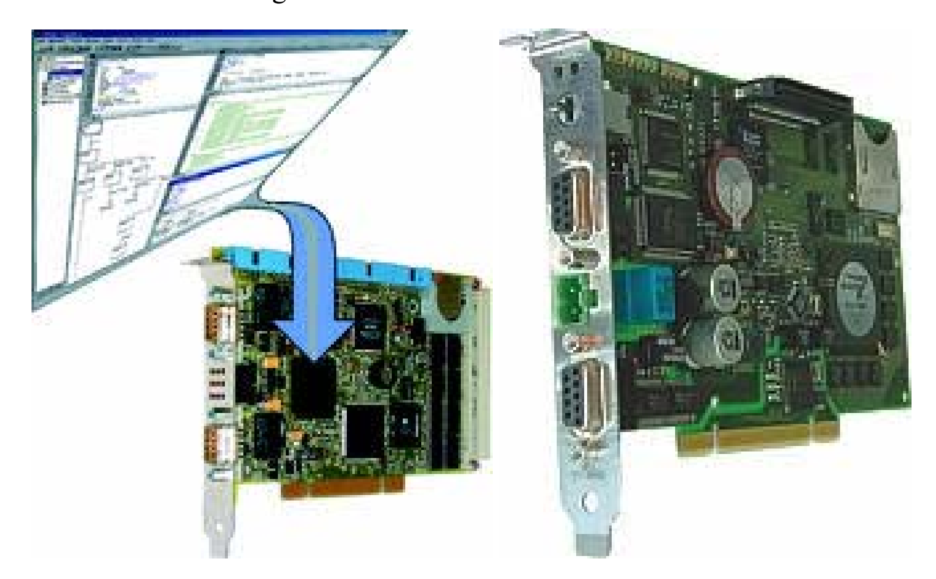


Abbildung 25: Slot-SPSen

Auf dieser Karte befindet sich ein Speichermodul, auf das sowohl SPS als auch PC zugreifen können. Durch diesen Speicher ist ein schneller Datenaustausch gewährleistet, wodurch diese SPS-Art die Echtzeitfähigkeit einer Schrank-SPS mit der Visualisierbarkeit und Editierbarkeit einer Soft-SPS vereint.

Die in dieser Diplomarbeit entwickelten Selbsttests können aufgrund der Hardwareähnlichkeit mit einer Schrank-SPS auch auf einer Slot-SPS eingesetzt werden. Es ist allerdings in solch einem Fall eine teilweise Neubeurteilung der ermittelten Diagnosedeckungsgrade der einzelnen Tests notwendig. Da die in Kapitel 10 beschriebenen Verfahren aufeinander aufbauen, muss die Neubeurteilung mit dem ersten Test begonnen werden, dessen Hardware sich von der der Schrank-SPS unterscheidet. Gleichfalls ist es notwendig einen neuen Test zur Überwachung des gemeinsam von PC und Slot-SPS genutzten Speicherbereichs zu entwickeln.

6.1.4 Sicherheits-SPS (SSPS)

In diesem Kapitel wird der Aufbau der SSPS „S5-115F“ von Siemens beschrieben, wie er im Artikel „Speicherprogrammierbare Steuerungen in der Sicherheitstechnik“ [7] dargestellt wurde.

Die Hardware der S5-115F ist aus zwei Standard-SPSen S5-115U (siehe Abbildung 26 – im weiteren als Teilgeräte bezeichnet) aufgebaut.

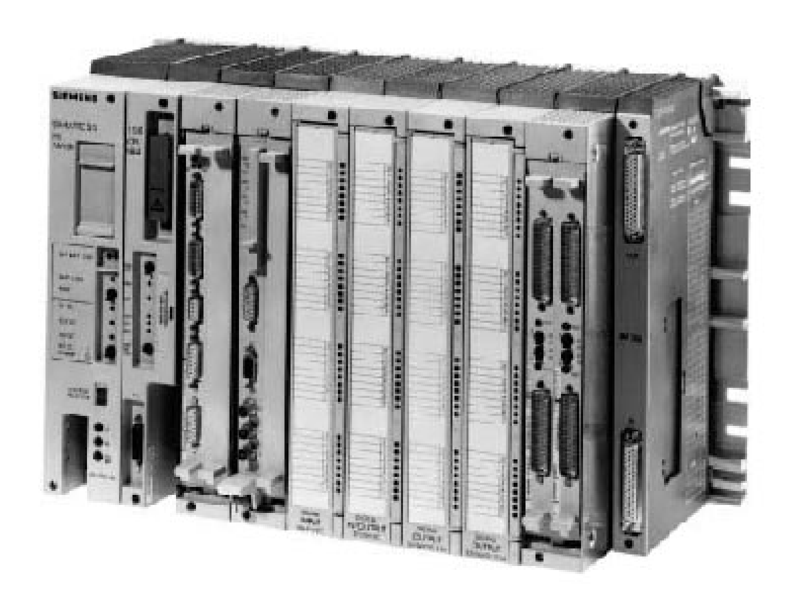


Abbildung 26: Standard-SPS S5-115U von Siemens

Beide Teilgeräte teilen sich zum Datenaustausch einen gemeinsamen Speicherbereich. Hierüber tauschen sie Daten zum Vergleich und zur Synchronisation aus. Zusätzlich laufen aus beiden Teilgeräten Selbsttest, die Fehler in ihrer Hardware entdecken sollen.

In jedem Zyklus werden zum Datenaustausch und zur Synchronisation:

- die Eingänge gelesen,
- das Speicherabbild der Eingänge mit dem anderen Teilgerät abgeglichen,
    - o dabei beide Teilgeräte synchronisiert,
- bei Gleichheit der von den Eingängen gelieferten Daten wird das Anwenderprogramm abgearbeitet,
- das Speicherabbild der Ausgänge abgeglichen,
- bei Gleichheit der von den Ausgängen gelieferten Daten werden die Ausgänge geschrieben.

Durch dieses Vorgehen können gefährliche Erstfehler entdeckt werden.

Neben einfachen Fehlern, die zum Abweichen der beiden Ausgangsergebnisse von einander führen und somit mit dem o. a. Verfahren erkannt werden, können in den SPSen Mehrfachfehler auftreten. Dies bedeutet, dass ein Fehler in dem einem der Geräte, der sich in dem zweiten Gerät wiederholt, bei beiden SPSen zum gleichen falschen Ergebnis an deren Ausgängen führt. Solche Mehrfachfehler werden durch den oben beschriebenen Abgleich der Daten von Ein- und Ausgängen nicht entdeckt. Durch Selbsttests der Teilgeräte sollen Mehrfachfehlern entdeckt werden.

Im Selbsttest werden dazu getestet:

- Prozessor
- RAM
- EEPROM

- Kopplung
- Programmablauf
- Peripheriebaugruppen

Diese Tests sind in der vom Anwender nicht beeinflussbaren Firmware der Teilgeräte installiert und werden stückweise in den Programmablauf der Teilgeräte integriert. Das bedeutet, dass ein Komplettest des Systems nicht in einem Zyklus sondern in mehreren aufeinander folgenden Zyklen abgeschlossen wird.

Abbildung 27 zeigt eine typische SSPS von PILZ.



Abbildung 27: Sicherheits-SPS von PILZ

Ohne den in diesem Kapitel beschriebenen Eingriff des Herstellers in die Hardware – Verbinden der Standard-SPSen über einen gemeinsamen Speicherbereich – ist dieses Konzept schwer zu realisieren (siehe Ausblick im Kapitel 13.3.6).

6.2 Hardware der Schrank SPS

Eine SPS arbeitet, wie jeder Computer, nach dem Prinzip der Eingabe, Verarbeitung und Ausgabe mit Speicher (EVA S-Prinzip – siehe Abbildung 28). Das bedeutet es gibt eine Möglichkeit Daten einzugeben, daraus Ergebnisse zu berechnen, zur weiteren Verarbeitung zu speichern und diese auszugeben.

Die Schrank-SPS besteht aus Prozessor, Speicher (RAM und EEPROM), und Ein- und Ausgängen. (siehe Abbildung 28)

Der Prozessor ist das „Verarbeitungszentrum" des SPS. Hier werden die Befehle und Daten aus dem Speicher mit den Daten, die die Eingänge liefern, verknüpft und an die Ausgänge weitergeleitet.

Im Prozessor sitzen unter anderem die Takt gebende Einheit (meist ein Quarz) und die ALU (Arithmetical Logical Unit – arithmetisch logische Einheit). Der Akkumulator (auch Akku genannt), der bei einer SPS durch das „Aktuelle Ergebnis" (AE) dargestellt wird und in dem die

Zwischenergebnisse gespeichert werden, befindet sich bei der SPS im normalen RAM-Speicher. (siehe zu AE Kapitel 6.3.2 und Kapitel 8.5)

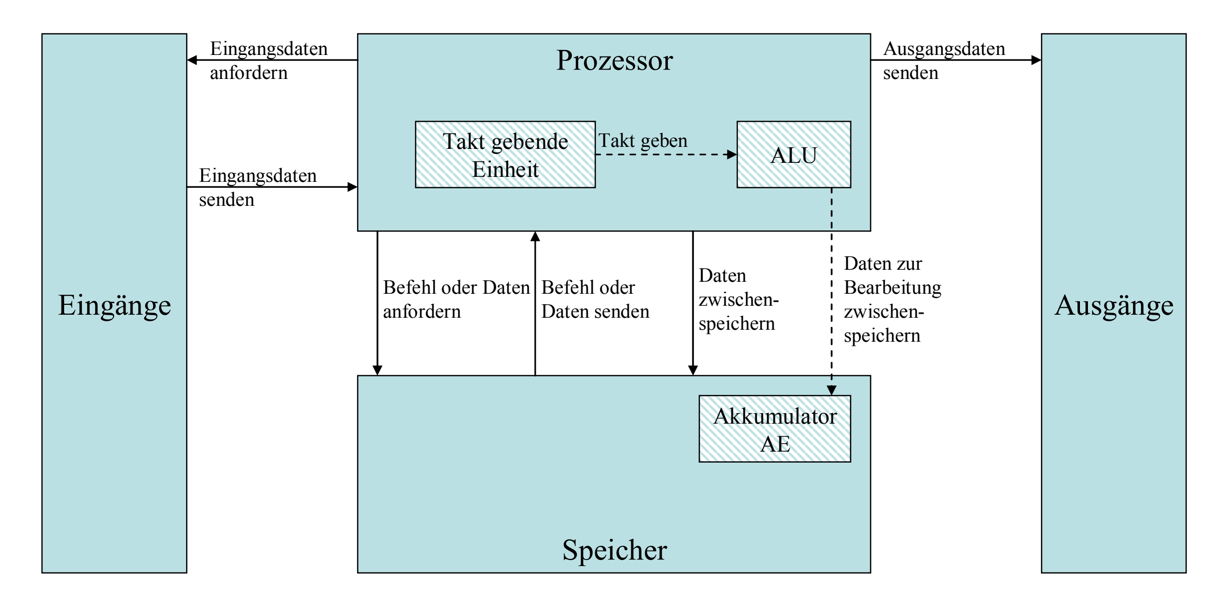


Abbildung 28: Prinzip der Eingabe, Verarbeitung und Ausgabe

6.2.1 Prozessor

Als Prozessoren werden überwiegend 32 bit RISC Prozessoren, zum Beispiel in der EH-WD10DR von Hitachi, eingesetzt.

RISC (Reduced Instruction Set Computing) bedeutet, dass dem Prozessor nur eine relativ eng begrenzte Anzahl an Befehlen zur Verfügung steht. Diese Befehle haben im Speicher alle die gleiche Länge.

Der Gegensatz zu RISC ist CISC (Complex Instruction Set Computing). Hier ist die Vielfalt der Befehle größer und die Länge der Befehle im Speicher variiert.

Die CISC Architektur wird in heutigen PCs eingesetzt. Für den begrenzten Funktionsumfang der SPS ist die RISC Architektur allerdings optimal. Durch die einheitliche Länge der Befehle eines RISC Prozessors können alle Befehle in einem Zyklus abgearbeitet werden, was die Geschwindigkeit der Verarbeitung erhöht. Dadurch kann die SPS niedrigere Taktraten fahren was zu weniger Verlustwärme führt. So kommen zum Beispiel SPS-Prozessoren der Firma SPEED7 nach Angabe des Herstellers [32] im von ihm angegebenen „normaler“ Industrietemperaturbereich (-20°C bis 60°C) ganz ohne Kühlung aus, da ihre Betriebstemperatur nur 15°C über der aktuellen Umgebungstemperatur liegen.

6.2.2 Speicher

SPSen verzichten in der Regel auf Festplatten und haben als Festspeicher statt dessen eine EEPROM Flash Karte. Die Größe der EEPROM Karten variiert zwischen 64 KByte und 4 MByte.

Da die Zugriffszeiten dieser Flash Karten verglichen mit anderen Speicherarten gering sind, wird das aktive Programm vor der Ausführung komplett in den flüchtigen Speicher, das RAM, übertragen. Die Größe des RAM liegt üblicherweise zwischen 16 KByte und 1 MByte.

6.2.3 Ein- und Ausgänge

Die Informationen aus diesem Kapitel sind dem Buch [13] entnommen.

Bei einer SPS verfügen die Eingangs- bzw. Ausgangskarten über mehrere Ein- bzw. Ausgänge. Die Eingangssignale verschiedener Eingänge werden dabei innerhalb der Eingangskarte zum Teil über unterschiedliche und zum Teil über dieselbe Hardware verarbeitet. (siehe Abbildung 29) Dies gilt analog auch für die auf der Ausgangskarte enthaltenen Ausgänge.

Ein einzelner Eingang einer Eingangskarte besteht aus:

- Anschlussklemmen für das Eingangssignal
- einem internen Relais
- einem Schalter
- einem RC-Filter
- einer galvanischen Trennung
- einem Schwellwertschalter
- einem Anschluss an den internen Bus der SPS

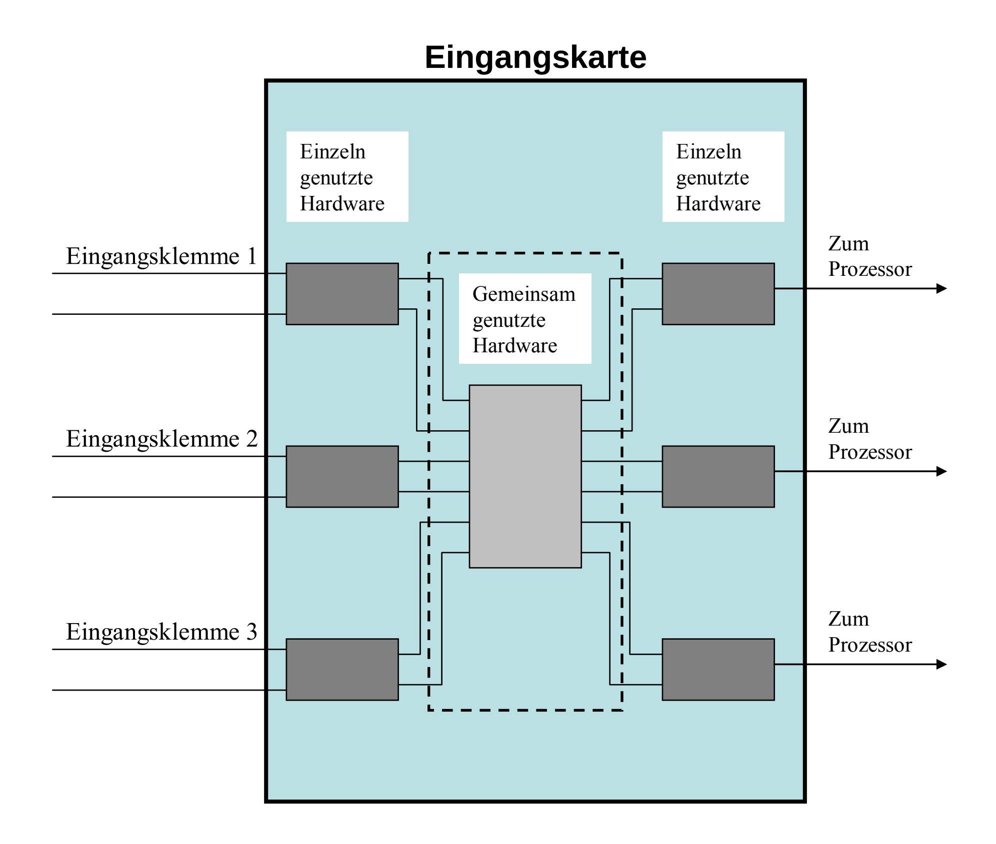


Abbildung 29: Signalweg durch eine Eingangskarte

Die Eingangskarte (siehe Abbildung 30) wird in der Regel mit 24 V Gleichstrom betrieben. Diese Spannung schließt, wenn sie an einem Eingang anliegt, über ein internes Relais einen Schalter.

Wenige SPSen lassen auch 230 V Wechselspannung am Eingang zu.

Intern arbeiten die Digitalen Ein- und Ausgangskarten mit 24 V.

Der durch das Relais betätigte Schalter schließt den Stromkreis der Versorgungsspannung (24 V). Diese 24 V fließen über ein RC-Filter und eine galvanische Trennung zu einem Schwellwertschalter.

Der Schwellwertschalter wandelt die Spannung in ein 0 V (keine Eingangsspannung) bzw. 5 V (Eingangsspannung vorhanden) Signal um, welches an den internen Bus weitergeleitet wird. Die Schwelle für den Schaltvorgang liegt im unteren Voltbereich um Spannungsschwankungen auszugleichen.

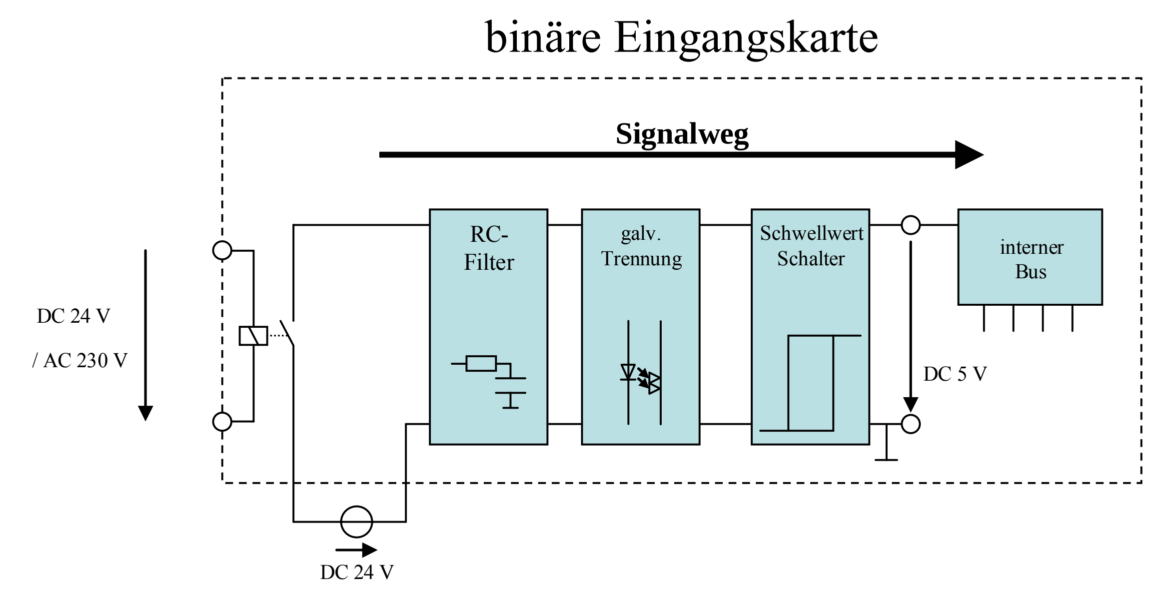


Abbildung 30: Eingang einer binäre Eingangskarte [13]

Bei der Ausgangkarte (siehe Abbildung 31) verläuft der Prozess entgegengesetzt. Der interne Bus gibt eine 0 V bzw. 5 V Spannung aus, die wiederum über eine galvanische Trennung an eine mit 24 V arbeitende Verstärker-Transistor Schaltung übergeben wird.

Da diese Schaltung nur Gleichstrom bis 0,5 A ausgeben kann besitzen einige Ausgangskarten Relaisausgänge, die auch Ströme bis 2 A schalten können.

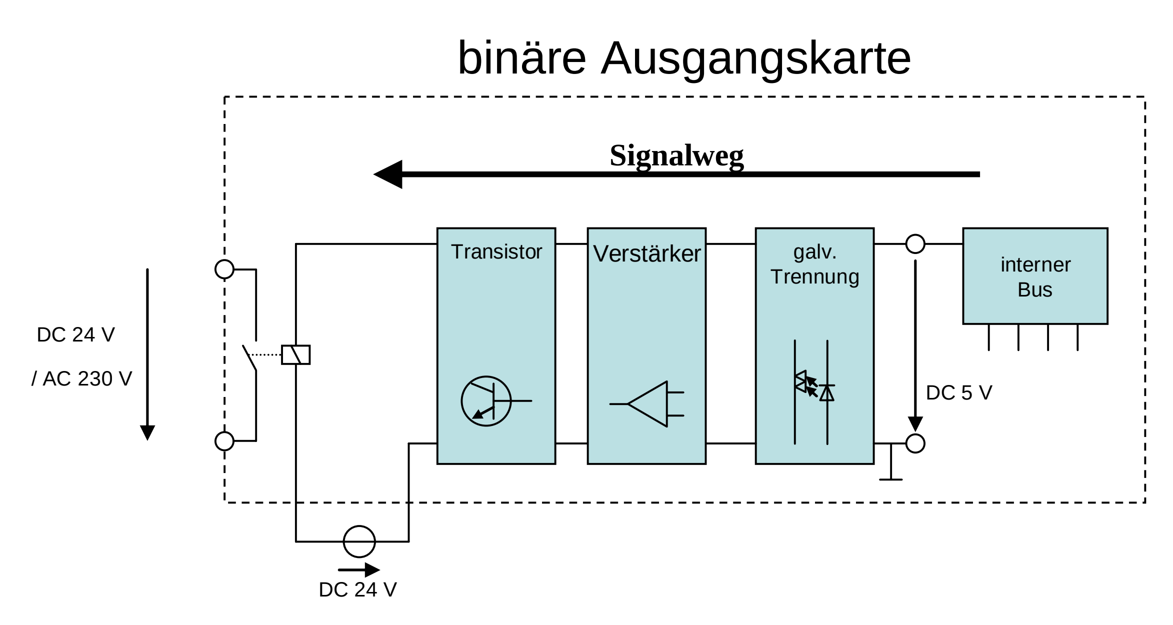


Abbildung 31: Ausgang einer binäre Ausgangskarte mit einem Relaisausgang [13]

6.3 SPS-Software

6.3.1 Zyklischer Ablauf

Das besondere bei einer SPS ist die zyklische Arbeitsweise. Wie Abbildung 32 darstellt, werden Eingabe, Verarbeitung und Ausgabe nach einander bearbeitet und nicht, wie z. B. bei einem PC, parallel. Veränderungen an den Eingängen, die nicht während eines Lesezyklus anliegen, werden nicht erfasst. Wird ein Ausgang im Laufe der Verarbeitung gesetzt und wieder zurückgesetzt, so ist am physikalischen Ausgang keine Änderung festzustellen. Hierdurch wird die Reaktionsgeschwindigkeit der SPS beeinflusst, was sich auf die in Kapitel 6.5 beschriebene Echtzeitfähigkeit auswirkt.

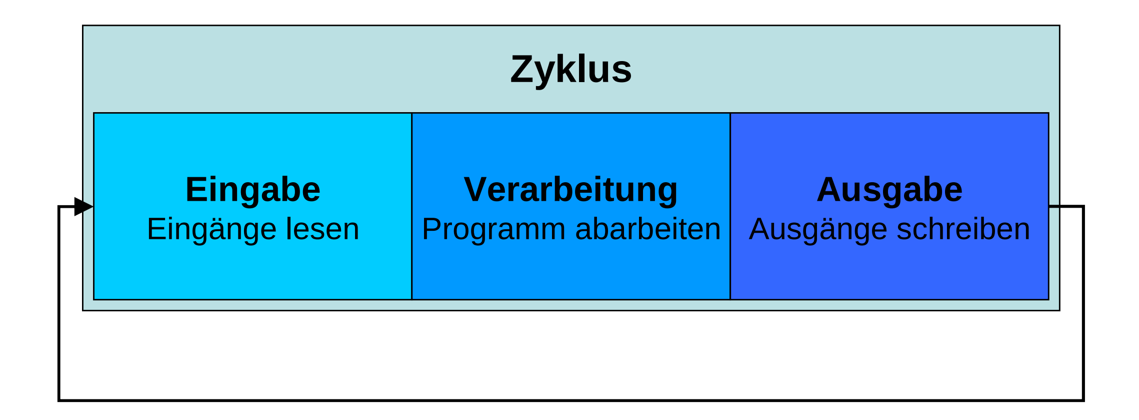


Abbildung 32: Prinzip der zyklischen Verarbeitung von Eingabe, Verarbeitung und Ausgabe

6.3.2 Aufteilung des Speichers

Das EEPROM beinhaltet das auszuführende Programm. Beim Starten der SPS wird der Programmcode des Programms vom langsamen EEPROM in das schnellere RAM geladen.

In Programmen, die mit der Programmierumgebung CoDeSys (siehe Kapitel 8.6) geschrieben sind, ist das RAM durch die Firmware der SPS in sechs Bereiche aufgeteilt (siehe Abbildung 33):

- Programmspeicher
- Speicherbereich für Globale Variablen
- Speicherbereich für frei verwendete Variablen
- Speicherbereich für Abbild der Ein- und Ausgänge
- Speicherbereich für speicherresistente Variablen
- Speicherbereich der das Aktuelle Ergebnis (AE) der letzten Oparation des Prozessors enthält

Zusätzlich ist im RAM auch das „Aktuelle Ergebnis“ (AE – siehe Kapitel 8.5) abgespeichert.

Programme sind zwar zur Laufzeit im RAM abgelegt, können aber selber nicht auf den Programmspeicher schreibend zugreifen. Es besteht allerdings keine physikalische Trennung wie bei der Harvard-Architektur, bei der Programmspeicher und Arbeitsspeicher physikalisch komplett von einander getrennt sind.

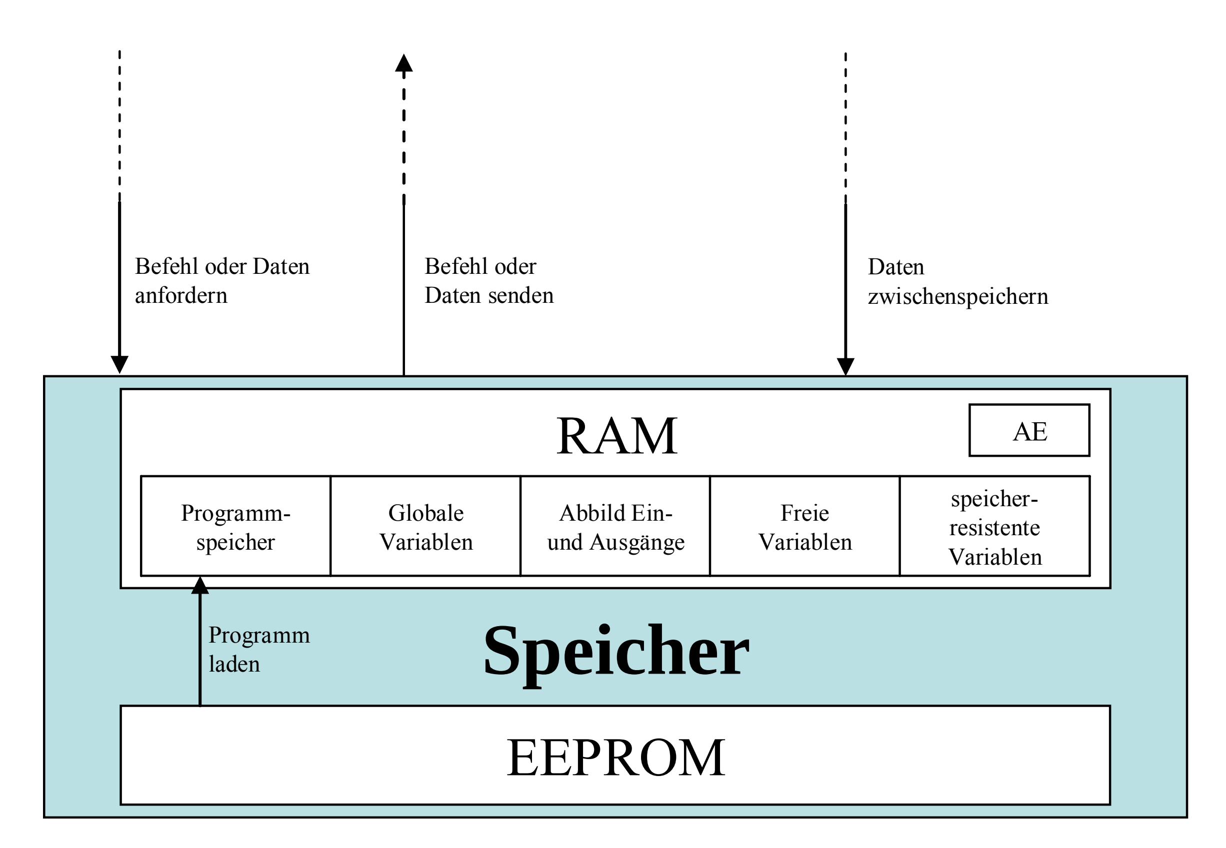


Abbildung 33: Aufteilung des Speichers aus Abbildung 28

6.4 Fehlererkennung in der SPS

Eine SPS ist im Allgemeinen durch den Hersteller mit einem Watchdog (siehe Kapitel 9.2.1 und 10.3.1) und einer CRC Prüfung des Programms (siehe Kapitel 10.5.1) ausgestattet. (siehe hierzu auch Kapitel 5.2.4)

Der Watchdog besitzt einen eigenen Takt, prüft aber nur gegen das Überschreiten einer vorgegebenen Zykluszeit.

Bei der CRC Prüfung wird das Programm nur beim Laden in den Speicher mit einer Prüfsumme verglichen, aber nicht während des zyklischen Betriebs.

6.5 Echtzeitfähigkeit

Soll eine Steuerung Sicherheitsfunktionen übernehmen, ist nicht nur die Ausfallsicherheit der Steuerung zu beachten, sondern auch deren Echtzeitfähigkeit. Echtzeitfähigkeit bedeutet, dass die Steuerung in einem angemessenen Zeitintervall auf Steuerungssignale reagieren kann. Sie hängt sowohl von der Reaktionszeit einer Steuerung als auch von der Geschwindigkeit des zu steuernden Prozesses ab. Während eine Steuerung bei der Steuerung einer Heizungsanlage echtzeitfähig sein kann, da nur jede Stunde ein Regelvorgang durchführt werden muss, kann dieselbe Steuerung bei der Steuerung eines Roboters zu langsam sein, da hier im Millisekundenbereich reagiert werden muss.

Da in dieser Diplomarbeit weder mit einer realen Schrank-SPS gearbeitet wurde, noch ein bestimmter Prozess betrachtet wurde, kann die Echtzeitfähigkeit einer SPS, die mit den beschriebenen Tests

erweitert wurde, nicht beurteilt werden. Dies muss bei der realen Umsetzung der Testprogramme im konkreten Einzelfall vom Programmierer der SPS beurteilt werden.

Die Reaktionsgeschwindigkeit der SPS hängt von der Zeit ab, die ein Signal benötigt, um von der Eingangskarte gelesen und von der Ausgangskarte geschrieben zu werden, sowie von der Zykluszeit des Programms, das auf der SPS abläuft.

7 Sicherheit einer SPS

In diesem Kapitel wird das Sicherheitsniveau der zur Zeit auf dem Markt erhältlichen Standard-SPSen und Sicherheits-SPSen bewertet und es wird aufgezeigt für welche sicherheitstechnischen Aufgaben diese SPSen verwendet werden können. Diese Ermittlung des Sicherheitsniveaus dient als Basis um die in dieser Diplomarbeit behandelten Standard-SPSen mit Diagnosefunktionen sicherheitstechnisch in diesem Umfeld einordnen zu können. Zur Bewertung der Standard-SPS mit Diagnosefunktionen siehe Kapitel 12.

7.1 Standard-SPS

Als Standard-SPS wird in dieser Diplomarbeit jede SPS (siehe Kapitel 6.1) außer einer vom Hersteller ausgewiesenen Sicherheits-SPS bezeichnet.

7.1.1 Einstufung der Standard-SPS nach DIN EN 954-1

Bereits durch das Auftreten eines einzelnen Fehlers in der Standard-SPS kann die Sicherheitsfunktion verloren gehen. Damit wäre die Standard-SPS nach der in Kapitel 5.2.1 vorgestellten DIN EN 954-1 Tabelle 2 (siehe Abbildung 7) in Kategorie B bzw. Kategorie 1 (bewährtes Bauteil) einzuordnen. Eine Standard-SPS gilt nach übereinstimmender Auffassung der Fachleute aber nicht als bewährtes Bauteil. Deshalb kann die Standard-SPS nach der vorgenannten Tabelle nur als „Kategorie B Bauteil" eingestuft werden. Wie der Risikograph dieser Norm in Abbildung 8 zeigt, kann die Standard-SPS deshalb in einer sicherheitstechnischen Steuerung nur zusammen mit zusätzlichen sicherheitstechnischen Maßnahmen eingesetzt werden.

7.1.2 Einstufung der Standard-SPS nach prEN ISO 13849-1

Standard-SPSen sind aus dem gleichen Grund, wie im vorherigen Kapitel beschrieben, in der prEN ISO 13849-1 der Kategorie B zuzuordnen. Die Bedingungen für die Einteilung der Steuerungen in Kategorien sind, wie in Kapitel 5.2.3 beschrieben, für die Kategorie B in der prEN ISO 13849-1 die gleichen wie für die Kategorie B in der DIN EN 954-1.

Da die SPS lediglich Kategorie B erreicht und außer dem Watchdog und der CRC-Prüfung (siehe hierzu Kapitel 6.4) keine sicherheitstechnischen Testeinrichtungen und somit auch keinen Diagnosedeckungsgrad besitzt, kann sie, abhängig von der $MTTF_d$ der einzelnen Komponenten, die Performance Level „a" und „b" erreichen. (siehe hierzu Tabelle 7 der prEN ISO 13849-1 in Abbildung 34)

7.1.3 Verwendungsmöglichkeiten der Standard-SPS

Nach dem Risikographen der DIN EN 954-1 in Abbildung 8 kann eine einzelne Standard-SPS unter zu Hilfenahme weiterer Maßnahmen für die sicherheitstechnische Steuerung von Maschinen eingesetzt werden, die entweder leichten Verletzungen hervorrufen können (Pfad S1), oder für solche, die schwere Verletzungen verursachen, wenn diese selten auftreten und vom Bediener verhindert werden können (Pfad S2 F1 P1).

Dieser Anwendungsbereich wird nach prEN ISO 13849-1 (siehe Abbildung 9) eingeschränkt. Hiernach kann eine Standard-SPS nur noch eingesetzt werden, wenn durch die Gefahrstelle an der Maschine lediglich leichte Verletzungen entstehen können, die selten auftreten (Pfad S1 F1). Hierbei sind zwei Fälle zu unterscheiden:

- Wenn diese Verletzungen vom Bediener verhindert werden können (Pfad S1 F1 P1) genügt eine Standard-SPS mit PL „a“.
- Sind diese Verletzungen nicht vom Bediener abzuwenden (Pfad S1 F1 P2), muss eine Standard-SPS des PL „b“ verwendet werden.

Im Gegensatz zur DIN EN 954-1 verlangt die prEN ISO 13849-1 in diesen Fällen allerdings keinen Einsatz zusätzlicher Maßnahmen.

7.2 Sicherheits-SPS

7.2.1 Einstufung der Sicherheits-SPS nach DIN EN 954-1

Die meisten Sicherheits-SPSen sind in der höchsten Kategorie 4 der DIN EN 954-1 zertifiziert, da sie nach ihrer Beschreibung in Kapitel 6.1.4 alle Vorraussetzungen der Tabelle 2 dieser Norm (siehe Abbildung 7) erfüllen.

Die wenigen Sicherheits-SPSen, die diese Kategorie nicht erfüllen, werden in dieser Diplomarbeit nicht weiter betrachtet.

7.2.2 Einstufung der Sicherheits-SPS nach prEN ISO 13849-1

Diese Sicherheits-SPSen der Kategorie 4 sind auch nach der prEN ISO 13849-1 der Kategorie 4 zuzuordnen. Die Bedingungen für die Einteilung der Steuerungen in Kategorien sind, wie in Kapitel 5.2.3 beschrieben, für die Kategorie 4 in der prEN ISO 13849-1 die gleichen wie für die Kategorie 4 in der DIN EN 954-1.

Neben der schon festgestellten Kategorie 4 hat eine solche Sicherheits-SPS die in Kapitel 6.1.4 beschriebenen Selbsttests mit hohem Diagnosedeckungsgrad und eine „hohe“ $MTTF_d$ (siehe Abbildung 17) der verwendeten Bauteile. Dadurch erreichen die am Markt verfügbaren Sicherheits-SPSen nach der in der nachfolgenden Abbildung 34 dargestellten Tabelle 7 aus der prEN ISO 13849-1 den höchstmöglichen Performance Level „e“.

Kategorie	B	1	2	2	3	3	4
DC_{avg}	keine	keine	niedrig	mittel	niedrig	mittel	hoch
$MTTF_d$ jedes Kanals: niedrig	a	Nicht abgedeckt	a	b	b	c	Nicht abgedeckt
$MTTF_d$ jedes Kanals: mittel	b	Nicht abgedeckt	b	c	c	d	Nicht abgedeckt
$MTTF_d$ jedes Kanals: hoch	Nicht abgedeckt	c	c	d	d	d	e

Standard-SPS

Sicherheits-SPS

Abbildung 34: prEN ISO 13849-1 Tabelle 7: Vereinfachte Bestimmung des Performance Levels [15] (siehe Abbildung 21) mit eingetragener Auswahl für Standard-SPS und Sicherheits-SPS

7.2.3 Verwendungsmöglichkeiten

Da die Sicherheits-SPS in beiden Normen den höchsten Sicherheitsstandard erfüllt, kann sie für jede sicherheitstechnische Steuerungsaufgabe eingesetzt werden. Der Nachteil beim Einsatz in niedrigeren Kategorien ist der relativ hohe Preis einer solchen Steuerung.

7.3 Schlussfolgerung

In diesem Kapitel wird anhand der aktuellen Normen belegt, warum die Standard-SPS wenigen bis gar keinen Sicherheitsanforderungen genügt und warum die Sicherheits-SPS die höchste Sicherheitseinstufung aufweist. (siehe hierzu Kapitel 1.2)

Das Einsatzgebiet der Standard-SPS im Bereich sicherheitstechnischer Steuerungen ist durch die Einstufung in die unterste Sicherheitsstufe (Kategorie 1 – Performance Level „a“) stark eingeschränkt und nach der heute gültigen DIN EN 954-1 nur unter Anwendung zusätzlicher sicherheitstechnischer Maßnahmen möglich. Die Sicherheits-SPS kann zwar in allen Bereichen der sicherheitstechnischen Steuerungen eingesetzt werden (Kategorie 4 – Performance Level „e“), ist aber, wie in der Einleitung erwähnt, durch ihren hohen Preis in den niedrigen Sicherheitsstufen unwirtschaftlich.

8 Programmierung einer SPS in AWL (nach DIN EN 61131-3)

Wie bereits in Kapitel 5.2.4 dargelegt, enthält die DIN EN 61131-3 die allgemeinen SPS Programmiersprachen. In diesem Kapitel wird die Programmierung einer SPS nach dieser Norm erläutert. Besonders wird hierbei auf die Sprache AWL eingegangen, die für die in dieser Diplomarbeit entwickelten Funktionsbausteine benutzt wurde.

8.1 Programmablauf

Aufgabe der SPS ist die in Abbildung 35 dargestellte Verarbeiten von ankommenden Signalen in abgehende Signale. Dieser Verarbeitung wird in der SPS mittels Programmbausteinen (siehe Kapitel 8.2.1) vorgenommen, die wiederum von so genannten „Tasks“ kontrolliert werden.

Der Programmierer muss im ersten Schritt die notwendigen Tasks generieren und diesen bestimmte Bedingungen zuweisen (Ereignis und Wertigkeit – s. u.), unter denen diese starten. Dies kann entweder der Eintritt eines Ereignisses (z. B. aufgrund des Öffnens einer Schutztür) sein, oder, bei einer so genannten freilaufenden Task, die freie Prozessorzeit wenn keine andere Task abläuft. Zusätzlich erhalten Tasks vom Programmierer Wertigkeiten zugewiesen, die beim gleichzeitigen Eintritt mehrerer Bedingungen die Reihenfolge des Task-Aufrufs regeln. Dies ist notwendig, da SPSen kein Multitasking beherrschen, also kein gleichzeitiges Abarbeiten mehrerer Tasks, und somit immer nur eine Task zur Zeit abläuft.

Dabei gilt, wie in Kapitel 6.3.1 beschrieben, die zyklische Abarbeitung des EVA Prinzips. D.h. es werden in einem Zyklus die Eingänge durch die SPS gelesen, danach die entsprechende Task abgearbeitet und hiernach die Ausgänge gesetzt.

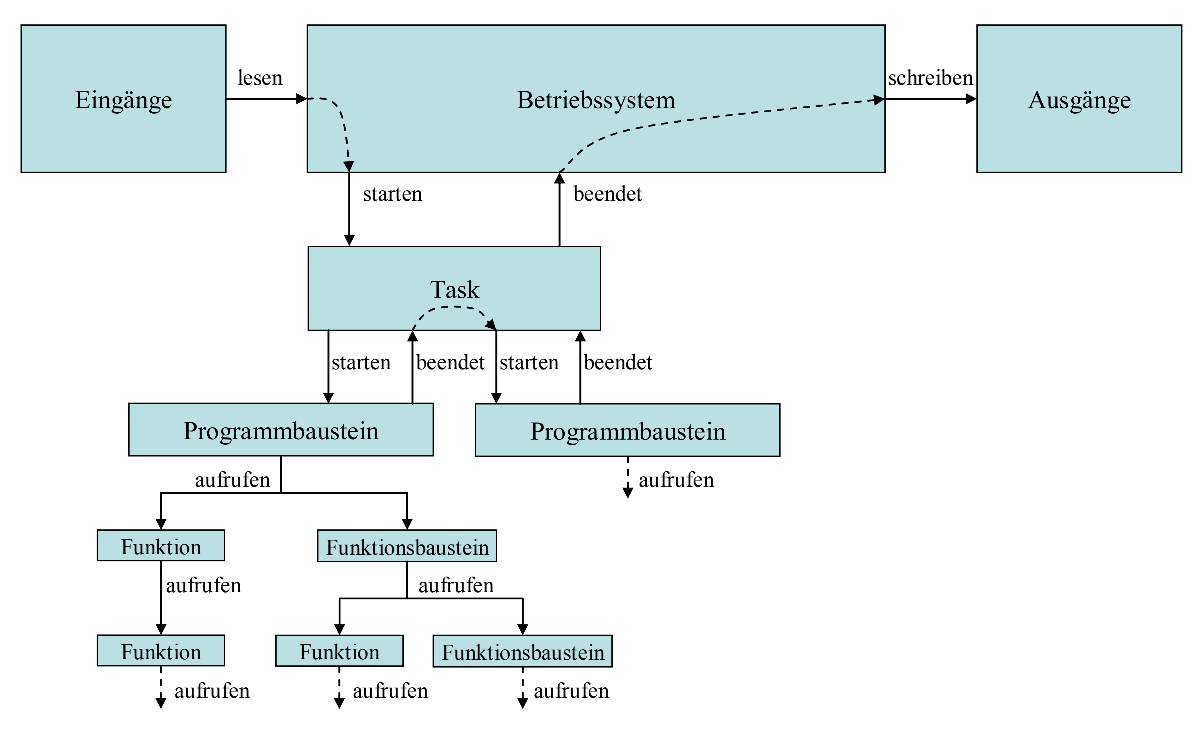


Abbildung 35: Task-Steuerung (Kapitel 8.1) und POEs (Kapitel 8.2)

8.2 Programmorganisationseinheiten

Nach der Norm werden Programme in Programmbausteine, Funktionsbausteine und Funktionen unterteilt. Diese werden als Programmorganisationseinheiten (POEs – siehe Abbildung 35) bezeichnet. Alle POEs können abzuarbeitenden Quelltext enthalten. Die Unterschiede in den Einzelnen POEs bestehen in ihren Rechten, Variablen zu deklarieren und in ihren Rechten andere POEs aufzurufen.

8.2.1 Programmbaustein

Der Programmbaustein kann verschiedene Arten von Variablen deklarieren und nutzen:

- Nur aus dem Programmbaustein heraus zugängliche Variablen (so genannte „lokale“ Variablen des Programmbausteins), bei der der Prozessor entscheidet welcher Speicherort gerade frei ist und diesen der Variablen zuweist
- Aus allen POEs zugängliche Variablen (so genannte „globale“ Variablen)
    - Variablen mit einer festen physikalischen Adresse, also einem genau definierten Bereich im Speicher.
    - Variablen, bei denen der Compiler zur Zeit der Übersetzung des Programms festlegt, welchen Speicherbereich sie zugewiesen bekommen.
    - Variablen, die im Speicher auch nach dem Abschalten der SPS erhalten bleiben (so genannte Retain-Variablen)

Der Speicherplatz einer lokalen Variablen eines Programmbausteins bleibt für diesen Programmbaustein reserviert, so dass der Programmbaustein lokale Variablen benutzen kann um sich Werte von einem Zyklus zu einem späteren Zyklus zu merken.

Ein Programmbaustein kann im Programmablauf Funktionsbausteine und Funktionen aufrufen aber keine anderen Programmbausteine.

8.2.2 Funktionsbaustein

Funktionsbausteine sind Unterprogramme zum Programmbaustein.

Sie können nur eigene lokale Variablen deklarieren.

Nutzen können sie:

- Globale Variablen
- Lokale, selbst deklarierte, Variablen
- Lokale Variablen aus POEs, aus denen sie aufgerufen werden (siehe Kapitel 8.4 – Pointer)

Der Speicherplatz einer lokalen Variablen eines Funktionsbausteins bleibt für diesen Funktionsbaustein reserviert, so dass der Funktionsbaustein diese Variablen benutzen kann um sich Werte von einem Aufruf zum nächsten, auch über mehrere Zyklen hinaus, zu merken.

Der Funktionsbaustein kann andere Funktionsbausteine und Funktionen aufrufen.

8.2.3 Funktion

Funktionen sind Berechnungshilfen. Sie können aus allen POEs heraus aufgerufen werden.

Ein Funktionsbaustein kann nur eigene lokale Variablen deklarieren und nutzen, die zwischen zwei Aufrufen nicht gespeichert werden. Aus diesem Grund liefert eine Funktion bei jedem Aufruf mit denselben Parametern immer dieselben Werte zurück.

Sie hat keinen Zugriff auf globale Variablen.

Funktionen können nur andere Funktionen aufrufen.

8.2.4 Rekursiver Aufruf

Ein rekursiver Aufruf ist, wenn ein Funktionsbaustein, oder eine Funktion sich selbst aufruft. Ist dieser Vorgang über mehrere Bausteine verteilt, wird es als indirekte Rekursion bezeichnet. In Abbildung 36 ist eine indirekte Rekursion über zwei Bausteine dargestellt, die sich gegenseitig aufrufen.

Ein rekursiver Aufruf der POEs, auch indirekt wie in Abbildung 36 am Beispiel zweier Funktionsbausteine dargestellt, ist nach der DIN EN 61131-3 [16] unzulässig. Das Einhalten dieser Bedingung wird vom Übersetzungsprogramm des Quelltextes (Compiler) überwacht.

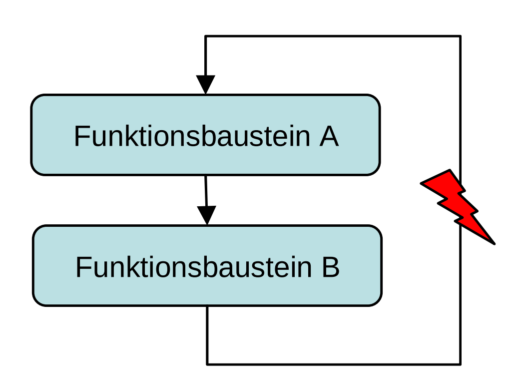


Abbildung 36: Beispiel eines indirekten rekursiven Aufrufs anhand zweier Funktionsbausteine

8.3 Programmiersprachen

Die Norm DIN EN 61131-3 [16] kennt fünf nachfolgend beschriebene Programmiersprachen, in denen der Anwender seine POEs schreiben kann. Davon sind zwei Text basiert und drei grafisch.

Ein Wechseln der Sprachen zwischen den einzelnen POEs ist möglich, so dass z.B. POEs, die schnell und effizient sein müssen, in Anweisungsliste (AWL – s. u.) geschrieben werden können und nachher mit anderen POEs in Funktionsbausteinsprache (FBS – s. u.) zusammengefügt werden können.

8.3.1 Anweisungsliste „AWL“ (Text basiert)

AWL ist die Assemblersprache für die SPS. AWL ist für die in dieser Diplomarbeit geschriebenen Tests als Sprache gewählt worden, da sie eine sehr Hardwarenahe und in der Ausführung schnelle Sprache ist.

Jede Zeile enthält einen Operator und einen Operanden. Fast jede programmierte Operation wird vom Compiler eins-zu-eins in Maschinencode übersetzt. Sprünge und Sprungmarken können beliebig

eingesetzt werden. Dies gibt die Möglichkeit Programme effizient zu programmieren d. h. unnötige Codezeilen zu vermeiden. AWL ist damit die Programmiersprache, die die schnellsten SPS Programme hervorbringt.

Die Syntax einer Zeile der Sprache besteht aus

[Sprungmarke]:	[Operator]	[Operand]	(*[Kommentar]*)

Dabei arbeitet AWL mit dem Aktuellen Ergebnis (AE), dem Akkumulator der SPS. Das AE sitzt im RAM der SPS (vergleiche Kapitel 6.2) und enthält das Ergebnis der zuletzt ausgeführten Operation. Es Gibt Operatoren, die das AE ändern, wie z. B. das Laden oder Addieren, und Operatoren, die das AE unberührt lassen, wie z. B. das Speichern. So ist es möglich Quelltext zu sparen, da z. B. Zwischenergebnisse nach dem Speichern nicht erneut geladen werden müssen.

Operatoren der Anweisungsliste (AWL) nach Tabelle 52 der Norm DIN EN 61131-3 [16]:

Nr.	Operator	Modifizierer	Bedeutung
1	LD	N	Setzt aktuelles Ergebnis dem Operanden gleich
2	ST	N	Speichert aktuelles Ergebnis auf die Operanden-Adresse
3	S R		Setzt booleschen Operator auf 1, falls das aktuelle Ergebnis boolesche 1 ist Setzt booleschen Operator auf 0 zurück, falls das aktuelle Ergebnis boolesche 1 ist
4	AND	N, (	Logisches UND
5	&	N, (	Logisches UND
6	OR	N, (	Logisches ODER
7	XOR	N, (	Logisches Exklusiv-ODER
7a	NOT		Logische Negation (Einer-Komplement)
8	ADD	(	Addition
9	SUB	(	Subtraktion
10	MUL	(	Multiplikation
11	DIV	(	Division
11a	MOD	(	Modulo-Division
12	GT	(	Vergleich: >
13	GE	(	Vergleich: >=

Nr.	Operator	Modifizierer	Bedeutung
14	EQ	(	Vergleich: =
15	NE	(	Vergleich: <>
16	LE	(	Vergleich: <=
17	LT	(	Vergleich: <
18	JMP	C, N	Sprung zur Marke
19	CAL	C, N	Aufruf Funktionsbaustein (siehe Tabelle 53)
20	RET	C, N	Rücksprung von aufgerufener Funktion, Funktionsbaustein oder Programm
21	)		Bearbeiten zurückgestellter Operation

Modifizierer werden dem Operator angehängt und verändern (modifizieren) damit seine Funktion.

Der Modifizierer „N“ negiert die Funktion des Operators. Z.B. lädt der Operator LDN das Komplement der angegebenen Speicherstelle in das AE.

Der Modifizierer „C“ führt dazu, dass der Operator nur dann ausgeführt wird, wenn das AE den Wert 1 hat. Z.B. ist der Operator JMPC ein Sprung der nur dann ausgeführt wird, wenn das AE=1 ist (ein so genannter „bedingter Sprung“).

Werden die Modifizierer „C“ und „N“ kombiniert, wird der Befehl nur dann ausgeführt, wenn das AE=0 ist.

Die geöffnete Klammer „(„ führt dazu, dass erst die Befehle in den Klammeroperatoren ausgeführt werden und danach der vorangestellte Operator.

Nachteile der Sprache liegen in den nicht vorhandenen Schleifen der Hochsprachen (z. B. „do ... while“) und deren Verzeigungen (z. B. „if-then-else“). Diese müssen durch bedingte Sprünge ersetzt werden. Durch viele Sprünge entsteht allerdings ein unübersichtlicher „Spagetticode“, der schlecht zu lesen und kaum zu warten ist. Programmteile in AWL sollten deshalb kurz gehalten und gut kommentiert werden.

8.3.2 Strukturierter Text „ST“ (Text basiert)

ST ist der Hochsprache PASCAL sehr ähnlich. Diese Sprache wurde erst spät in die Norm aufgenommen. Seit diesem Zeitpunkt wurde sie allerdings in vielen neueren SPSen und Programmiersystemen implementiert, da die Hersteller auf die Entwicklung einer eigenen Hochsprache verzichtet haben. Die meisten SPS Hersteller halten in den anderen Bereichen aber an ihren alten Sprachen fest.

Programmieraufgaben sind in ST kompakter darstellbar als in AWL, wodurch sie besser lesbar werden. Andererseits produziert ST bei der Lösung gleicher Aufgaben mehr Maschinencode als AWL, wodurch die benötigte Zeit der Ausführung eines Programms steigt.

ST ist hinsichtlich der Effizienz zwischen AWL und den grafischen Programmiersprachen anzusiedeln.

8.3.3 Funktionsbausteinsprache „FBS“ (grafisch)

Die FBS, auch Funktionsplan (FUP) genannt, ordnet Funktionen und Funktionsbausteine wie in einem Blockschaltbild oder Logikplan an. Dadurch werden vor allem binäre Logikschaltungen leicht verständlich. Damit bietet diese Sprache einen leichten Einstieg und ist deshalb für den ungeübten Anwender geeignet.

FBS wird nach Aussage des Herstellers von CoDeSys (3S) hauptsächlich in europäischen Betrieben verwendet.

8.3.4 Kontaktplan „KOP“ (grafisch)

Der KOP ist an die Darstellung von Schaltplänen angelehnt. Das Programmieren geschieht durch das grafische Verbinden von Objekten.

Diese Art der Programmierung wird nach Aussage des Herstellers von CoDeSys (3S) vielfach im amerikanischen Raum eingesetzt.

8.3.5 Ablaufsprache „AS“ (grafisch)

In AS lassen sich besonders gut sequentielle Aufgaben programmieren.

Die einzelnen Schritte des Programms sind als Kette aufgereiht, wobei zwischen den Schritten jeweils eine Weiterschaltbedingung angebracht ist. Ist diese erfüllt, springt das Programm in den nächsten Schritt.

Mit dieser Programmiersprache können vor allem Prozessabläufe gut programmiert werden, die nach einander ablaufen.

8.4 Variablendeklaration

Die Art der Deklaration der Variablen ist unabhängig von der gewählten Programmiersprache. Sie läuft in folgenden Schritten ab:

- Startbefehl zur Deklaration

  „VAR“

- Auswahl der Art der Variablen (lokal, global, retain… – siehe z. B. Kapitel 8.2.1)

  z. B. Lokal = „“, global = „_GLOBAL“, retain = „_RETAIN“, …

- Auswahl eines Variablen-Namens (beliebige Länge, keine vom Programm belegten Bezeichnungen, nur bestimmte Zeichen)

  z. B. „a“, „b_c“, „d-e“, …

- Festlegung der Speicherstelle (fest oder programmgesteuert)

  z. B. „AT %MX5.7“ als feste Zuordnung auf das 7te Bit des 5ten Bytes des Bereichs der globalen Variablen des Speichers. Wird hier keine Festlegung getroffen, steuert das Programm die Speicherbelegung selbst.

- Zuordnung eines Variablen-Typs (Bool, Byte, Integer, Word, Doubleword…)

  z. B. „: BOOL;“, „: BYTE;“, „: INT;“, „: WORD;“, „: DWORD;“,…

- Abschlussbefehl zur Beendung Deklaration

  Immer „END_VAR“

Ein gesamter Aufruf könnte zum Beispiel sein:

VAR_GLOBAL

Hilfs_Variable AT %MX4.4 : BOOL;

END_VAR

Damit würde der Speicherstelle des 4ten Bits des 4ten Bytes eine boolsche globale Variable mit Namen „Hilfs_Variable“ zugeordnet.

Eine feste physikalische Adresse kann mehreren Variablen zugeordnet sein. Dies macht grundsätzlich Doppelbelegungen möglich. Allerdings muss beachtet werden, dass ungewollte Doppelbelegungen zu Programmfehlern führen, weshalb die direkte Adressierung beim Programmieren möglichst vermieden werden sollte. (siehe Abbildung 37)

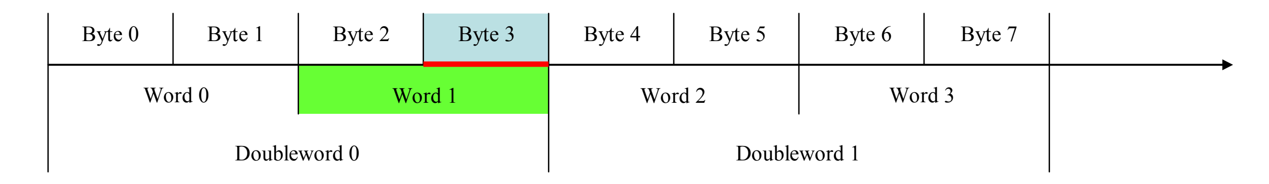


Abbildung 37: Doppelbelegung des Speichers – dargestellt am Zahlenstrahl

Doppelbelegungen passieren in der Praxis, wie in Abbildung 37 gezeigt, häufig durch die Verwendung unterschiedlicher Variablentypen. Dies liegt daran, dass durch die unterschiedliche Länge der Variablen-Typen die Übersicht über die belegten Speicherplätze verloren geht. Jeder Typ beginnt am Anfang des Speichers mit Null zu zählen und teilt hiernach den gesamten Speicher in Schritte seiner Länge auf. So ist das dritte „Byte“ des Speichers gleichzeitig die zweite Hälfte des ersten „Word“ (ein Word = 2 Byte). In diesem Beispiel wird eine Variable (grün) angelegt, die das erste Word anspricht und eine Variable (blau), die das dritte Byte anspricht. Eine Veränderung der Variable im ersten Word löst damit auch eine Veränderung des Wertes der Variablen im dritten Byte aus. Doppelbelegungen können auch vom gleichen Datentyp sein. Dies kann aber eher erkannt werden.

Eine Adressierung von Variablen über Zeiger (Pointer) ist nach Norm generell nicht vorgesehen. Die Norm ermöglicht jedoch einem Funktionsbaustein einen Pointer auf eine Variable zu übergeben. Die Referenz dieses Pointers, d. h. die Speicherstelle auf die der Pointer zeigt, ist allerdings vom Funktionsbaustein selbst nicht veränderbar. Das heißt, sie kann vom Funktionsbaustein nicht auf eine andere Speicherstelle umgelenkt werden und wird beim Programmieren wie eine normale Variable benutzt. Allerdings ist es dem Funktionsbaustein dadurch möglich, den Inhalt der Variablen des ihn aufrufenden POEs zu verändern. Darüber kann ein Funktionsbaustein auf lokale Variablen anderer POEs zugreifen.

8.5 Unterschiede zu „normalen“ Programmierumgebungen

Die DIN EN 61131-3 ist darauf ausgelegt, das Programmieren möglichst einfach zu halten. Das bedeutet, es werden nur die Funktionen angeboten, die zum Programmieren einer technischen Anlage oder Maschine benötigt werden. Die Einschränkungen gegenüber „normalen“ Programmierumgebungen sind hierbei insbesondere:

- Es gibt keine Pointer, außer den in Kapitel 8.4 beschriebenen.
- Es gibt keine Statusbits, die z.B. einen Überlauf bei einer Rechenoperation anzeigen.
- Im Unterschied zum normalen Prozessor hat eine SPS keinen Akkumulator, einen Prozessoreigenen Speicher, in dem das Ergebnis der letzten Operation gespeichert wird. Der Prozessor der SPS speichert das aktuelle Ergebnis (AE) in einer freien Speicherstelle des RAMs ab. (siehe Abbildung 28)
- Rekursive Aufrufe sind, wie in Kapitel 8.2.4 beschrieben, unzulässig.
- Das Abfragen des Programmzählers ist nicht möglich.
- Keine der in Kapitel 8.3 beschriebenen Sprachen ist Objekt orientiert. Das bedeutet es gibt keine Vererbung von Eigenschaften oder Variablen zwischen einzelnen Programmorganisationseinheiten.

Diese Einschränkungen wurden gemacht, um den Programmcode einfach und damit auch sicherer zu gestalten. Mehr Funktionalität birgt auch mehr Möglichkeit, Fehler zu machen.

8.6 CoDeSys

Der Name der Programmierumgebung CoDeSys leitet sich von „Code Development System“ ab. Hiermit wurden die Tests aus Kapitel 10 programmiert und getestet.

Der Markt für SPSen wird von Siemens und vor allem von der Siemens-SPS „Simatic S7“ beherrscht.

3S (Smart Software Solutions) initiierte im Jahr 2000 die Gründung einer Vereinigung mehrerer SPS Hersteller. Unter dem Namen „CoDeSys Automation Alliance“ (CAA) haben sich bis Mai 2006 insgesamt 78 Hersteller zusammengeschlossen. 57 dieser Hersteller, also die überwiegende Mehrheit, kommt aus Deutschland. Siemens gehört dieser Vereinigung nicht an.

8.6.1 Gründe für den Einsatz von CoDeSys in dieser Diplomarbeit

Der wichtigste Grund für die Entscheidung im Rahmen dieser Diplomarbeit CoDeSys zu benutzen und damit gegen die Siemens Programmierumgebung, war die Entscheidung, die Tests in der Programmiersprache AWL (siehe Kapitel 8.3.1) und damit in einer in der Norm DIN EN 61131-3 [16] beschrieben Standardprogrammiersprache zu verfassen.

Siemens unterstützt in seiner Umgebung diese Normsprache nicht.

Zusätzlich zum Befehlsumfang von AWL hat CoDeSys eigene Operatoren. (siehe Kapitel 8.6.2) Diese wurden nur in einem der Tests, der in Kapitel 10.6.3 beschrieben wird, benutzt. Alle anderen Tests sind normkonform programmiert und sind damit in allen normkonformen Umgebungen einsetzbar.

Hinzu kommt, dass CoDeSys eine gute Möglichkeit bietet, Programme in der Soft-SPS (siehe Kapitel 6.1.2) von CoDeSys ablaufen zu lassen und hier mit einem Debugger auf Fehler zu überprüfen. Dies wurde genutzt um die Tests wie in Kapitel 10.1.4 beschrieben auf deren richtige Funktion hin zu überprüfen.

8.6.2 Zusatzfunktionen in CoDeSys

CoDeSys bietet dem Programmierer die Möglichkeit, neben den Funktionen der Norm, Pointer einzusetzen und Speicherstellen auch in Funktionsblöcken direkt zu adressieren.

Zum Unterstützen der Arbeit mit Pointern dient ein Befehl, mit dem die Adresse einer Variablen festgestellt werden kann.

In CoDeSys ist es auch möglich Variablen in Funktionsbausteinen direkt zu adressieren. (siehe Kapitel 8.2.2)

9 Fehler erkennende Programme

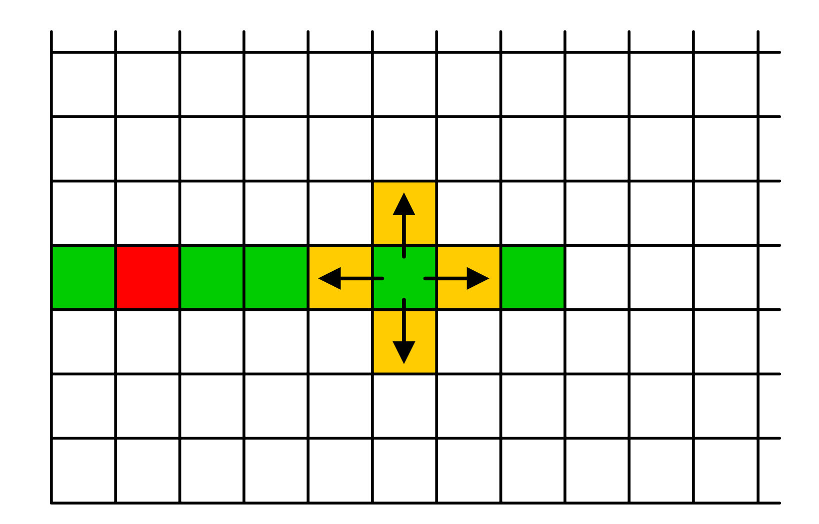
Fehler in einer SPS können sowohl während der Programmierung als auch während des Betriebs entstehen. Während Programmierfehler durch organisatorische Maßnahmen vermieden bzw. gemindert werden können, müssen zur Erkennung und Beherrschung von zufällig auftretenden Hardwarefehlern Testprogramme und Hardwareredundanzen eingesetzt werden.

Die „möglichen Fehler“ (F) aus Kapitel 9.1, die durch Fehler in der Hardware einer SPS ausgelöst werden können, sowie die „allgemeinen Lösungen“ (L) aus Kapitel 9.2 und die „programmierten Test“ (T) in Kapitel 10 sind in diesen Kapiteln mit den Bezeichnungen Fx, Lx bzw. Tx (x steht für die Nummern/Bezeichnungen des Fehlers) versehen. Dies soll die gegenseitige Zuordnung erleichtern. Eine Gesamtübersicht findet sich der Tabelle in Kapitel 11.

9.1 Fehler-Möglichkeiten durch Hardware-Fehler einer SPS

Grundsätzlich können in jedem Teil der SPS Fehler auftreten:

- F1) Die Takt gebende Einheit (meist ein Quarz) kann zu schnell sein oder zu langsam. Ein Ausfallen der Einheit kann dabei als extreme Form von „zu langsam“ angesehen werden. (siehe hierzu auch [4])
- F2) Im Prozessor kann der Programmzähler einen „Stuck at“ Fehler haben. Hierbei kann eine Zelle eines Speichers, hier des Speichers des Programmzählers, nicht mehr beschrieben werden und gibt beim Lesen dauerhaft eine Null bzw. eine Eins zurück. (siehe Abbildung 38)
- F3) Im Speicher kann das AE des Prozessors einen „Stuck at“ Fehler (s. o.) haben. (siehe Abbildung 38)
- F4) Die ALU (Arithmetical Logical Unit) des Prozessors kann Fehler bei der Ausführung von logischen oder arithmetischen Befehlen produzieren. Auch kann es zur Ausführung falscher Befehle durch Fehler im Interpreter der ALU kommen.
- F5) Die Kommunikation zwischen Prozessor und Speicher kann gestört sein, so dass es zum Schreiben und / oder Lesen falscher Daten kommt.
- F6) Eine Zelle im Speicher kann einen „Stuck at“ (s. o.) Fehler haben. (siehe Abbildung 38)
- F7) Im Speicher kann ein Setzen bzw. Rücksetzen einer Speicherstelle durch das Setzen oder Rücksetzen einer anderen, in der Regel benachbarten, Zelle geschehen. Dies ist ein so genannter Übertragungsfehler. (siehe Abbildung 38)
- F8) Bei den Ein- und Ausgangskarten kann das Ausfallen ihrer Komponenten (Kapitel 6.2.3) zum dauerhaften Lesen bzw. Ausgeben einer Null oder Eins führen. Besonders gefährdet ist hier das eingebaute Relais, das „kleben bleiben“ kann (dauerhaft Eins durch Verschmelzen der Kontakte im Relais).



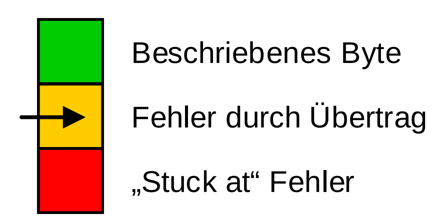


Abbildung 38: Speicherbild mit Fehlermöglichkeiten

Damit kommt die Diplomarbeit hinsichtlich der notwendigen Testverfahren auf das gleiche Ergebnis für die zu prüfenden Bauteile wie der Hersteller der in Kapitel 6.1.4 beschriebene Sicherheits-SPS. Die programmierten Tests aus Kapitel 10 sind ohne den in Kapitel 6.1.4 beschriebenen Eingriff in die Firmware erfolgt. Sie beschränken sich nach der Aufgabenstellung der Diplomarbeit auf die reine Anwender Ebene. Durch die hieraus entstehenden Einschränkungen liegt die erreichbare Sicherheitsstufe deshalb unterhalb der einer Sicherheits-SPS.

9.2 Strategien der Fehler-Entdeckung

Mittels Selbsttest ist es meist schwierig einen konkreten Fehler eines bestimmten Bauteils zu finden. Es ist jedoch teilweise möglich deren Auswirkungen im System festzustellen. Nachfolgend werden die in der Industrie allgemein üblichen Strategien zur Feststellung von Systemfehlern durch Selbsttests erläutert. Diese sind wie in Kapitel 9 beschrieben geordnet.

9.2.1 L1) Takt

Der vorhandene Takt (siehe Abbildung 28) kann mittels eines Watchdogs überprüft werden, der prüft, ob das Programm in einer bestimmten Zeit abgearbeitet wird. Dies kann aber nur geschehen, wenn der Watchdog eine eigene Taktquelle nutzt. Ein solcher „externer Watchdog“ kann meist nur das Überschreiten einer festgelegten Zeit feststellen. Bessere Ergebnisse werden erzielt, wenn der Watchdog mit einem definierten Zeitfenster arbeitet, da dann auch ein Unterschreiten der vorgegebenen Zeit festgestellt wird.

9.2.2 L2) Programmzähler

Ein Programmzähler ist das Register im Prozessor, in dem die Adresse des aktuellen Befehls abgelegt ist. Nach dem Abarbeiten eines Befehls wird der Programmzähler um einen definierten Wert erhöht (der Länge eines Befehlssatzes), sofern er nicht vom Befehl selbst (z. B. einem Sprungbefehl) verändert wurde.

Fehler im Programmzähler können mittels einer Ablaufüberwachung festgestellt werden. Die hier aufgezeigten Ablaufkontrollen lassen sich in der Literaturquelle [4] nachlesen.

Eine Ablaufüberwachung, wie in Abbildung 39 dargestellt, besteht aus einer Testroutine, die während des Programmablaufs regelmäßig aufgerufen wird. Wird die Testroutine an einer definierten Stelle durch einen Fehler übersprungen oder mehr als einmal aufgerufen, so stellt die Ablaufüberwachung dies fest. Je enger dabei die Prüfungen aneinander liegen, desto wahrscheinlicher wird die Entdeckung des Fehlers.

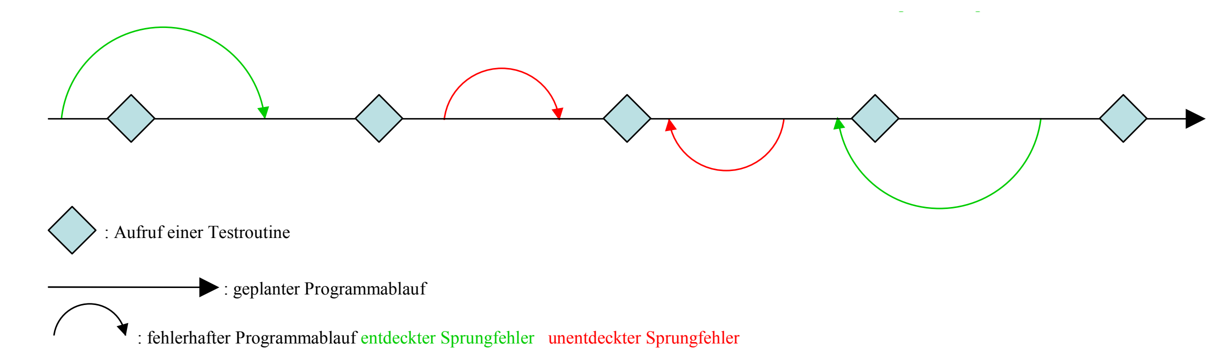


Abbildung 39: Entdeckte und unentdeckte zufällige Sprünge

Ein Zweiter Ansatz, Fehler im Programmzähler festzustellen ist die Zeitliche Überwachung, die in Abbildung 38 dargestellt ist. Macht ein Programm einen unbeabsichtigten Sprung so ändert sich seine Laufzeit. Dieser Sprung kann sowohl die Laufzeit verkürzen (überspringen von Programmbefehlen) als auch verlängern (mehrfache Ausführung von Programmbefehlen – Schleife). Die Testroutine prüft in diesem Fall, ob sie zum richtigen Zeitpunkt aufgerufen wurde. Um die Toleranz beim Zeitmessen, die nicht aus Fehlern resultiert, zu berücksichtigen wird ein definiertes Zeitfenster benutzt. Je enger das Zeitfenster ist, um so eher wird ein fehlerhafter Sprung erkannt.

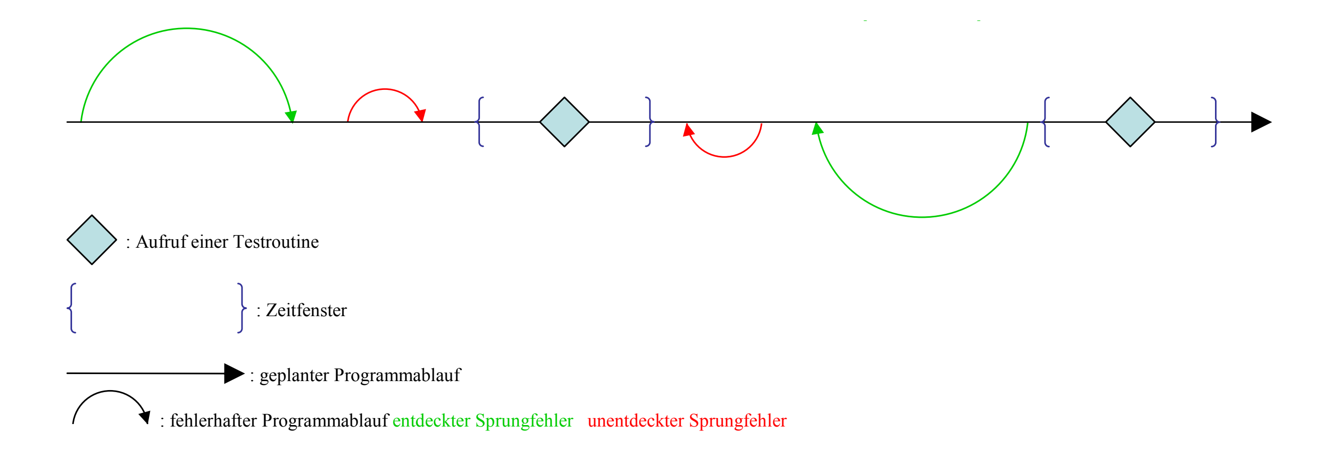


Abbildung 40: Erkennung fehlerhafter Sprünge durch ein Zeitfenster

Wird die Ablaufüberwachung mit der zeitlichen Überwachung kombiniert, erhöht dies den Grad der Fehleraufdeckung. Eine Darstellung dieser Kombination findet sich in Abbildung 41.

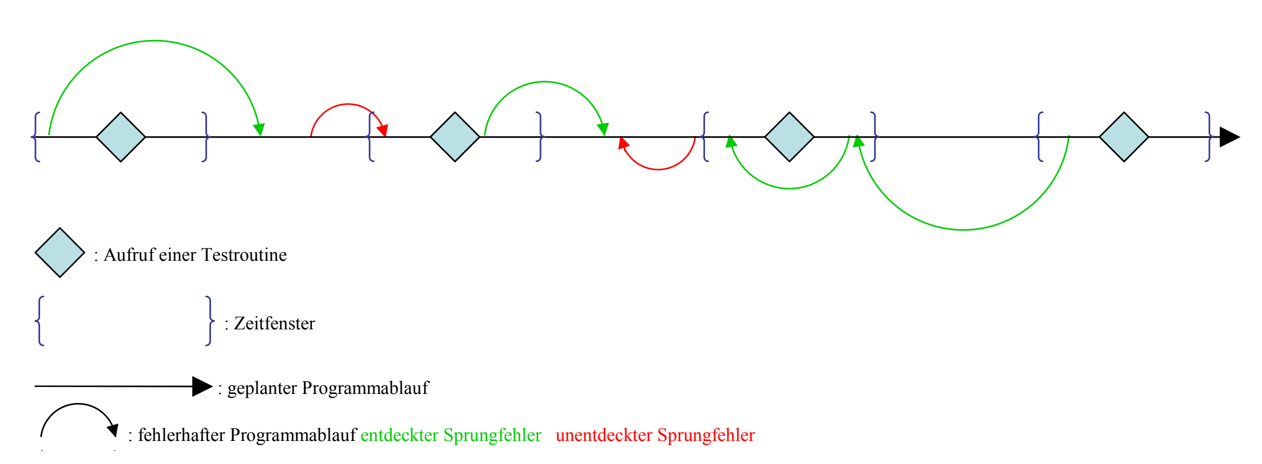


Abbildung 41: Ablaufkontrolle durch kombinierte Überwachung

9.2.3 L3) Akkumulator

Fehler im Akkumulator (siehe Abbildung 28) können mittels eines einfachen „walking Bit" Tests erkannt werden. Dabei wird eine Eins vom niederwertigsten Bit bis zum höchstwertigsten Bit „geschoben" und dabei überprüft, wann das Bit gesetzt wird, dass einen Übertrag anzeigt. Eine Umsetzung dieses Tests findet sich im BGIA Report [1].

9.2.4 L4) Befehlsausführung

Zum Überprüfen der Befehlsausführung in der ALU des Prozessors (siehe Abbildung 28) werden Ergebnisse berechnet, und mit erwarteten Prüfsummen verglichen, bzw. bedingte Sprünge durchgeführt und deren Sprungziel geprüft. Auch hierzu finden sich Grundlagen der Umsetzung im BGIA Report [1].

9.2.5 L5) Kommunikation zwischen Speicher und Prozessor

Eine im BGIA Report [1] vorgestellte Methode zur Prüfung des Kommunikationswege ist das wiederholte Lesen und Speichern von unterschiedlichen Zahlen, sowie die Überprüfung, ob die gespeicherte Zahl mit der gelesenen übereinstimmt.

9.2.6 L6+7) Speicher

Für den Test des Speichers gibt es zahlreiche Methoden. Zwei dieser Möglichkeiten werden nachfolgend dargestellt:

L6) Die einfachste Methode besteht darin die benutzten Speicherstellen nach einander zu beschreiben, wieder auszulesen, und zu prüfen, ob beim Lesen der Speicherstelle der gleiche Wert gelesen wird, der vorher geschrieben wurde. Hierbei werden häufig bestimmte Bit-Muster eingesetzt. (siehe Abbildung 53 ff)

L7) Die rechnerisch und damit auch zeitlich aufwändigste Methode besteht darin, den ganzen Speicher bitweise zu beschreiben und nach jedem Schreibvorgang, neben dem Erfolg des Schreibens, jedes andere Bit im Speicher auf eine Veränderung hin zu prüfen. Diese Überprüfung wird häufig eingesetzt, um defekte Speicherstellen zu finden, wenn ein Fehlverhalten des Speichers vorliegt.

Weitere Methoden und deren Zuverlässigkeit sind in der Norm DIN EN 61508 [17] aufgezeigt.

9.2.7 L8) Ein- und Ausgänge

Eingänge können nur mittels Redundanz, also dem Einlesen eines Wertes über zwei verschiedene Eingänge und dem Vergleich der an den Prozessor weitergeleiteten Werte, getestet werden.

Ausgänge können nur mittels Rücklesen der ausgegebenen Werte durch einen Eingang getestet werden. Dazu wird der Ausgang mit dem Eingang mittels externer Verschaltung verbunden. Dadurch liegt vor dem Eingang das gleiche Signal wie hinter dem Ausgang an. Bei fehlerfreien Ein- und Ausgangskarten muss dadurch das Signal vor dem Ausgang mit dem Signal hinter dem Eingang übereinstimmen. Dies wird durch den Prozessor überprüft.

Die Ausfälle in einem Ausgang stellen sich meist durch das „kleben bleiben“ auf einem Wert (ein oder aus) dar. Diese Ausfälle würden durch den vorgenannten Test erst in dem Moment bemerkt, in dem der Ausgang geschaltet wird. Da dieser Ausfall nicht erst bemerkt werden darf, sobald das Bauteil benutzt wird, müssen Ausgänge in regelmäßigen Zeitabschnitten ein- und ausgeschaltet und diese Funktion geprüft werden. Dies muss auch möglich sein, wenn das laufende Programm den Ausgang anders nutzen will als die Testroutine. Bei einem Mikrocontroller kann der Ausgang kurzeitig abgeschaltet und wieder einschaltet werden, bevor ein nach geschaltetes Relais dies mitbekommt. Ein entsprechender Test ist in dem BGIA Report [1] programmiert. Im Falle der SPS ist dies, wegen der eigenen Relais und des zyklischen Programmablaufs aufgrund des damit verbundenen Geschwindigkeitsverlustes nicht möglich. Dieses Problem wurde im Rahmen der Diplomarbeit untersucht und eine Lösung entwickelt. (siehe Kapitel 10.7.3)

10 Standard SPS mit Fehler erkennenden Programmbausteinen

10.1 Allgemeines

Wie schon in den Kapiteln 2.1, 6 und 8.6 erwähnt, wurden alle in diesem Kapitel dargestellten Tests in der Programmierumgebung CoDeSys geschrieben und wie in Kapitel 10.1.4 dargestellt auch in dieser Umgebung auf ihre Funktion hin überprüft.

Im Anschluss an die Erläuterung des jeweiligen Testalgorithmus und dessen Darstellung im Ablaufdiagramm wird der Diagnosedeckungsgrad des vorgestellten Tests nach DIN EN 13849-1 [15] (siehe Kapitel 5.2.3) bestimmt. Die ermittelten Diagnosedeckungsgrade werden in Kapitel 12.2 zur Berechnung des Performance Levels einer Standard-SPS mit Diagnosefunktion benutzt.

10.1.1 Auswahl der Programmiersprache

Aus den beschriebenen Programmiersprachen in Kapitel 8.3 wurde wie hier begründet AWL ausgewählt, da es bei den einzelnen Tests um Schnelligkeit in der Abwicklung und die Kontrolle über den erzeugten Maschinencode geht.

10.1.2 Verbesserte Fehlererkennung durch programmtechnische Zusatzfunktionen

Die in diesem Kapitel aufgezeigten Testprogramme sollen die vorhandene – stark eingeschränkte – Fehlererkennung (siehe Kapitel 6.4) einer Standard-SPS unterstützen und erweitern. Die Tests bauen in der Reihenfolge aufeinander auf, wie sie in diesem Kapitel beschrieben werden. Das bedeutet, dass spätere Tests Bauteile und Funktionen der SPS benutzen, die von früheren Tests überprüft werden. Es ist z. B. nicht ratsam einen Speichertest durchzuführen, ohne vorher zu überprüfen ob der Vergleichsoperator, der den geschriebenen mit dem gelesenen Speicherinhalt vergleicht, fehlerfrei funktioniert.

Ein Implementieren eines Tests ohne die ihm vorzuschaltenden Tests kann dazu führen, dass ein Fehler, trotz Überwachung unerkannt bleibt.

10.1.3 Implementieren der Tests in SPS-Programme

Die in dieser Diplomarbeit beschriebenen Tests sind keine Programmbausteine, die einfach zu einem bestehenden Programm hinzugefügt werden können. Obwohl alle Tests im Hinblick auf einfache Integration gestaltet wurden, muss der Programmierer der SPS sich mit den Tests auseinandersetzen um diese in sein Programm zu integrieren.

Die zu erledigende Aufgabe des Programmierers liegt bei einigen Tests nur in der zeitlichen Planung der Testzyklen. Bei anderen Tests, wie zum Beispiel der Ablaufkontrolle in Kapitel 10.3.2, muss der Programmierer diese an verschiedenen, auf den konkreten Einzelfall abgestellten Stellen in sein gesamtes Programm integrieren.

Die Quellcodes aller Tests dieser Diplomarbeit befinden sich in Anhang II der Diplomarbeit, der auf der der Diplomarbeit beigefügten CD zu finden ist.

10.1.4 Verifikation und Validierung der erarbeiteten Testverfahren

Alle in diesem Kapitel vorgestellten Testverfahren wurden im Einzelschritt-Modus des Debuggers von CoDeSys auf ihre Funktion überprüft. Der Einzelschritt-Modus erlaubt das genaue Verfolgen des Programmablaufs. In diesem Modus ist es außerdem jederzeit möglich Variablen beliebige Werte zuzuweisen, wodurch bestimmte Fehlerzustände simmuliert werden können.

Beim Speichertest wurden in diesem Modus Fehler in den überwachten Speicherstellen simuliert, indem zwischen der ersten und zweiten Signaturbildung (siehe Kapitel 10.6.2.1) Werte in überwachten Speicherstellen geändert wurden bzw. die Werte der geprüften Speicherstelle selbst geändert wurden. Die Reaktion auf diese Fehler wurde danach im Einzelschritt-Modus überprüft.

Benutzt ein Test eine Variable mehrfach wie z. B. der Test des Speichern und Ladens (Kapitel 10.4.2.6), so wurde die Variable stets im Einzelschritt-Modus zur Laufzeit modifiziert.

Anstatt die fehlerhaften Werte im Einzelschritt-Modus zur Laufzeit zu implementieren können diese auch direkt im Quellcode des Tests fehlerhaft implementiert werden, wenn sie, wie beim Test der bedingten Sprungbefehle (Kapitel 10.4.2.1), nur einmal geladen werden. In diesem Fall ist es für die Validierung nicht relevant ob der korrekte Wert geladen und dann modifiziert, oder direkt modifiziert geladen wird.

Die Validierung des Tests der Ein- und Ausgänge (Kapitel 10.7.3) wurde durchgeführt, indem die Eingangsvariable, die dem im Test geprüften Ausgang zugewiesen war, nach einigen Zyklen gesetzt, bzw. nicht gesetzt wurde. Damit konnte die externe Hardwareverschaltung an der im Rahmen der Diplomarbeit verwendeten Soft-SPS simuliert werden.

10.1.5 Bewertung der Diagnosedeckungsgrade (DCs) der Testverfahren

Die Bewertung von Diagnosedeckungsgraden in dieser Arbeit ist an die Bewertungen ähnlicher Tests aus den Normen DIN EN 61508 und prEN ISO 13849-1 angelehnt. In diesen Normen sind die Testverfahren in drei Kategorien aufgeteilt:

- „niedrig“ mit 60% – < 90% DC
- „mittlere“ mit 90% – < 99% DC

  und
- „hohe“ mit >=99% DC

Ein Test, der 100% aller möglichen Fehler entdeckt ist laut diesen Normen nicht möglich. Soweit die Testbewertungen auf der Grundlage eines Vergleich mit den Tests aus den o. .a Normen basieren, wurden die o. a. Werte übernommen.

Abweichend davon konnten bei einigen Testverfahren Berechnungsgrundlagen gefunden werden, die eine genaue Angabe des DC erlauben. In einigen Fällen war eine genaue Berechnung trotz Berechnungsgrundlage nicht möglich. Die hier ermittelten „Wertespannen“ wurden deshalb auf die „glatten“ Werte 60% bzw. 90% abgerundet.

10.2 Reaktion auf einen erkannten Fehler

Die sicherste Methode eine SPS nach einem erkannten Fehler abzuschalten ist es einen Sprungbefehl auf sich selbst zeigen zu lassen. (siehe Abbildung 42) Die dadurch erzeugte Endlosschleife löst nach kurzer Zeit den Watchdog der SPS aus.

```
Error:
JMP Error
```

Abbildung 42: Allgemein übliche Reaktion auf einen erkannten Fehler

Es besteht aber die Möglichkeit, dass ein schnelleres Abschalten nötig ist, als es der Watchdog zulässt, oder das der Watchdog selbst einen Fehler hat und es dadurch zu gar keinem Abschalten kommt. Ist eine zweite SPS, wie ab Kapitel 10.8 erläutert, zur Überwachung zugeschaltet, würde diese im Fall des Nichtabschaltens den Fehler sehen und ihrerseits, allerdings erst einige Zyklen später, abschalten.

In dieser Diplomarbeit wird im Fehlerfall ein Baustein aufgerufen, der vor der o. a. erzwungenen Schleife weitere Befehle ausführen kann. So kann z. B. der Speicherbereich, der die Ausgangswerte der Ausgänge der SPS enthält, vor der Schleife in einen definierten Zustand (im Bespiel Null) versetzt werden. Zusätzlich wird eine speicherresistente Variable *Fehler_bemerkt* gesetzt, die im Fehlerfall ein erneutes Starten durch den Bediener verhindert. (siehe Abbildung 44)

Ein möglicher Fehler beim Funktionsaufruf wurde bedacht, weshalb auch im aufrufenden Funktionsbaustein eine Endlosschleife programmiert ist. (siehe Abbildung 43)

Beliebiger Testbaustein:

```
[…]
Error:
CAL FB_ERROR                 (*Aufruf des Fehlerbausteins*)
Cal_Error:
JMP Cal_Error                (*Schleife im Falle des Versagens des CAL Befehls*)
```

Abbildung 43: Aufruf des Bausteins FB_ERROR in einem beliebigen Testbaustein

FB FB_ERROR

```
LD 0
ST Output                    (*Alle Ausgänge direkt abschalten*)
LD TRUE
ST Fehler_bemerkt
Error:
JMP Error
```

Abbildung 44: Quellcode des Bausteins FB_ERROR

10.3 Fehlererkennung durch Ablaufkontrolle

10.3.1 T1a) Standard Watchdog Funktion

10.3.1.1 Erläuterung

Die vom Hersteller eingebaute Watchdog Funktion der SPS überwacht das laufende Programm auf das Überschreiten einer Zykluszeit. Der Watchdog ist also nur ein rückwärts laufender Timer, der am Ende eines Zyklus geladen (getriggert) wird.

Der Watchdog schaltet die SPS ab, sobald die Programmabarbeitung langsamer wird, stoppt oder wenn das Programm in einer Endlosschleife gefangen ist.

Je nach SPS ist der Watchdog vom Hersteller gegen unbeabsichtigtes Triggern geschützt. In dem Fall wird er nicht durch ein einzelnes Signal zurückgesetzt, sondern durch eine bestimmte Signalfolge.

Eine direkte Möglichkeit den Watchdog zu überprüfen gibt es nicht, da jedes Auslösen der Schutzfunktion die SPS anhält und eine Überprüfung ohne den Watchdog auszulösen nicht möglich ist.

10.3.1.2 Diagnosedeckungsgrad

Eine Ablaufkontrolle, zu der der Watchdog zählt, wird in der Literatur [4] nach fünf Punkten beurteilt.

Fehleraufdeckung

- beim Programmzähler
- bei Endlosschleifen
- bei Veränderung der Taktfrequenz
- beim Ausfall der Stromversorgung
- beim Ausfall des Prozessors

Da die meisten Watchdogs einer SPS keine eigene Stromversorgung besitzen, ist die Fehleraufdeckung bei Ausfall der Stromversorgung gleich Null.

Endlos Schleifen und ein Ausfall des Prozessors werden zu 100% erkannt.

Da der Watchdog nur eine Verlängerung der Zykluszeit, und somit eine Verlangsamung der Taktfrequenz, aber kein Verkürzung der Zykluszeit erkennt, liegt die Fehleraufdeckung hier bei annähernd 50%. Dieser Wert ist zusätzlich abhängig vom Spielraum zwischen der Durchlaufzeit eines Zyklus und der Auslösezeit des Watchdog, da eine Verlängerung der Zykluszeit um diesen Zeitraum ebenfalls nicht erkannt wird.

Ein Fehler im Programmzähler wird nur in den seltensten Fällen erkannt. Ein Wert zwischen 5% und 10% ist hier laut Literaturquelle [4] realistisch.

10.3.2 T2a) Selbst programmierte Watchdog Funktion mit Ablaufkontrolle

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	ABLAUF_UEBERWACHUNG
Namen der Nebenfunktionen	ABLAUF_UEBERWACHUNG_OHNE_ZEIT ABLAUF_UEBERWACHUNG_RESET ABLAUF_UEBERWACHUNG_SCHLEIFE
Kurzbeschreibung	Kontrolle des Programmablaufs auf unerwartete Sprünge
Eingänge	
GeringsterVorgaenger	Die Bausteinnummer die mindestens vor dem aktuellen Testbaustein aufgerufen werden musste.
Programmstelle	Die Bausteinnummer des aktuellen Testbausteins.
minZeit	Zeit die mindestens seit der letzten Zeitmessung vergangen sein muss
maxZeit	Zeit die maximal seit der letzten Zeitmessung vergangen sein darf
Ausgänge	
keine	
Genutzte globale Variablen	
LetzteProgrammstelle	Speichert die letzte aufgerufene Bausteinnummer aller angegebenen Testbausteine.
LetzterPruefzeitpunkt	Speichert den Zeitpunkt des letzten Aufrufs von ABLAUF_UEBERWACHUNG oder ABLAUF_UEBERWACHUNG_RESET.

10.3.2.1 Erläuterung

Ein selbst programmierter Watchdog kann nie einen Fehler im eigenen Takt der SPS bemerken, da er diesen selbst für seine Funktion nutzt.

Darum hat der hier angegebene Watchdog nur die Aufgabe den Ablauf des Programms zu überwachen und nicht den Takt. Dazu nutzt er die Kombination der in Kapitel 9.2 genannten Verfahren. (siehe Abbildung 45)

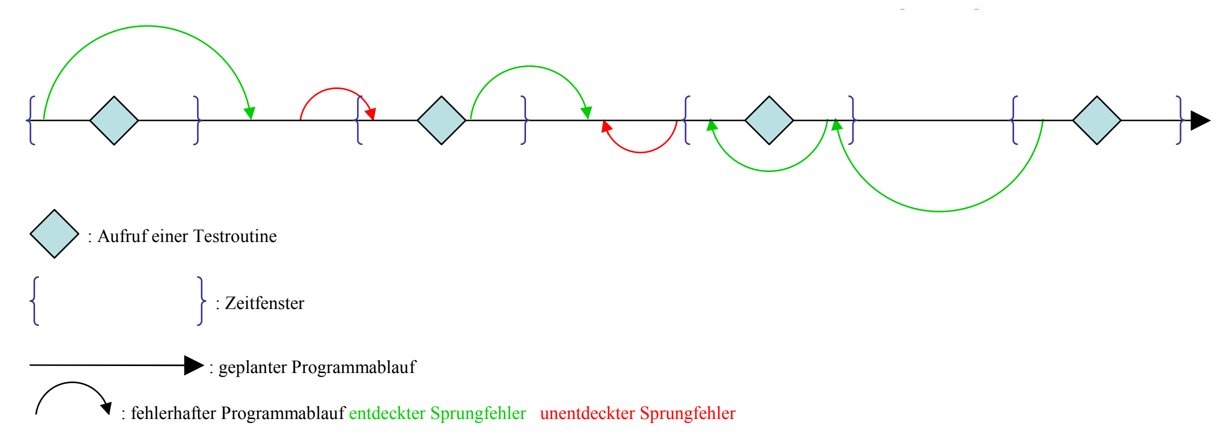


Abbildung 45: Ablaufkontrolle durch kombinierte Überwachung (siehe Seite 74)

Damit die einzelnen Tests funktionieren, muss der Programmierer über die Variablen *minZeit* und *maxZeit* festlegen, wie viel Zeit seit dem Aufruf des letzten Tests vergangen sein sollte. Zusätzlich muss über die Variable *GeringsterVorgaenger* festgelegt werden welcher Test der Vorgängertest ist. Dies kann bei einer Schleife oder Programmverzweigung allerdings kompliziert sein, da in diesem Fall der Vorgänger variieren kann.

In der Literaturquelle [4] wird mit verschiedenen Verfahren zum Aufsummieren und Prüfen von Signaturen gearbeitet. Eine Signatur ist die mathematische Verknüpfung von Zahlen, die aufgrund ihrer einzelnen Werte oder der Art der Verknüpfung ein möglichst eineindeutiges Ergebnis bilden. Durch den Vergleich dieses berechneten Wertes mit einem festgelegten oder vorher berechneten Wert kann die Signatur überprüft werden. (für weitere Informationen zu Signaturen siehe Kapitel 10.6.2.1) Die Verfahren die mit Signaturprüfungen arbeiten haben alle den Nachteil, dass sie entweder ungenau in der Fehlererkennung, oder schwierig in der Anwendung sind.

Das nachfolgend beschriebene Verfahren ist dagegen relativ einfach anzuwenden, und bringt trotzdem ein hohes Maß an Sicherheit.

Im Gegensatz zu den in der Literaturquelle [4] beschriebenen Verfahren wird kein Unterschied zwischen Signatur setzenden und Signatur prüfenden Funktionen gemacht. Stattdessen wird eine globale Zählervariable benutzt, die in jedem Funktionsaufruf überprüft und neu gesetzt wird. Dadurch entfallen aufwändige Berechnungen von Prüfsummen seitens des Programmierers. Durch das dadurch ständige Abprüfen des Programmablaufs und dem engen zeitlichen Rahmen der bei jedem Testaufruf durch den Programmierer vorgegeben werden kann, ist trotzdem ein hohes Maß an Sicherheit gegeben. Um auch Quersprünge zwischen parallel ablaufenden Programmzweigen festzustellen, ist es ratsam in diesen Zweigen verschiedene Testnummern zu vergeben.

Abbildung 46 zeit einen Beispielablauf:

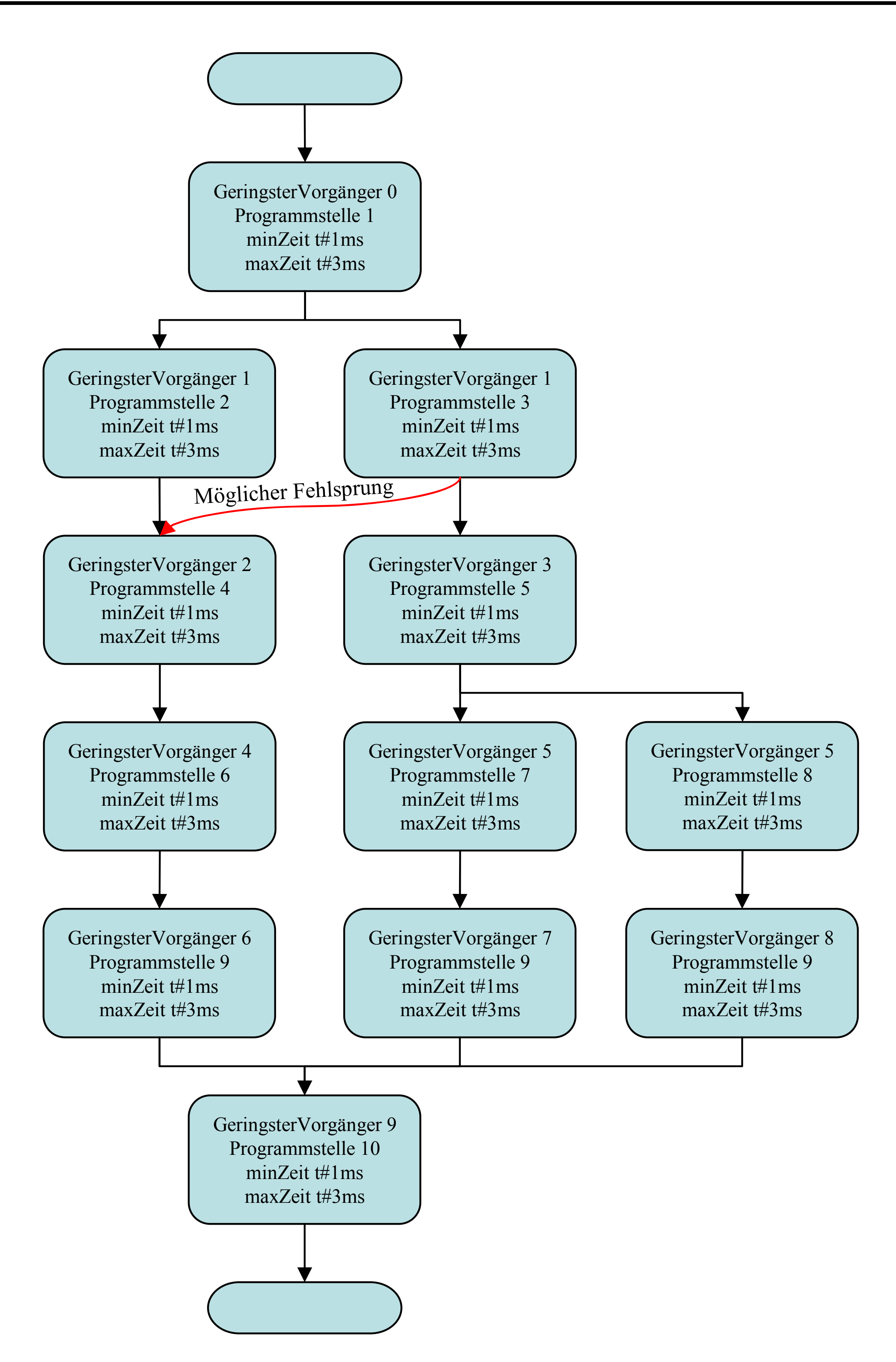


Abbildung 46: Beispielaufruf für die programmierte Ablaufüberwachung

Das Programm hätte in diesem Fall jeweils ein bis drei Millisekunden Zeit von einem Aufruf eines Testbausteins bis zum Nächsten zu gelangen. Ein Unterschreiten oder ein Überschreiten der Zeit bis zum Aufruf des nächsten Testbausteins, führt beim Aufruf des nächsten Testbausteins zu einem Fehler.

Die Bausteine sind durchnummeriert, so dass die Nummern auch in verschiedenen Zweigen möglichst dicht aufeinander folgen. Dies ist nötig um möglichst wenige unentdeckte Ablauffehler zuzulassen. So kann nach dem Baustein mit der $Programmstelle\ 3$ ein möglicher Fehlsprung in den linken Zweig zwischen $Programmstelle\ 2$ und $Programmstelle\ 4$ nicht entdeckt werden, da er die Bedingung $GeringsterVorgänger\ 2$ des Bausteins mit der $Programmstelle\ 4$ erfüllt. Ein Quersprung aus dem linken Zweig von $Programmstelle\ 2$ aus hinter diesen Baustein hingegen wird entdeckt. Die Wahrscheinlichkeit eines solchen unentdeckten Fehlsprungs ist allerdings gering, da es in allen Fällen in der Regel nur wenige Stellen gibt, die fehlerhaft angesprungen werden können, ohne das der Fehlsprung entdeckt wird. Die letzten Bausteine in den unterschiedlichen Zweigen sollten alle die gleiche Nummer tragen (in diesem Fall die $Programmstelle$ 9). Dadurch wird ein Sprung am Ende eines Teilzweiges in einen anderen Teilzweig aufgedeckt.

Neben dem Hauptbaustein ABLAUF_UEBERWACHUNG existiert der Baustein ABLAUF_UEBERWACHUNG_OHNE_ZEIT. Dieser arbeitet, wie in Abbildung 47 dargestellt, ähnlich dem Hauptbaustein, mit der Ausnahme, dass er keine Zeit misst. Beide müssen in einem sinnvollen Verhältnis zueinander verwendet werden.

Zur Ermittlung des Verhältnisses in dem die beiden Bausteine zu verwenden sind, muss die Dauer des Programmzyklus betrachtet werden. In der Simulationsumgebung von CoDeSys ist der Watchdog fest auf 100 Millisekunden eingestellt. Dies ist hier die maximale Dauer eines Zyklus. Da die interne Uhr von CoDeSys eine Auflösung von einer Millisekunde hat, bringt eine Anzahl von Testbaustein-Aufrufen von größer hundert keine Verbesserung mehr, da nur hundert Zeitabschnitte zu messen sind. Beim Einsatz der Testverfahren sollten in diesem Fall 10 Testverfahren mit Zeiterkennung benutzt werden und der Rest ohne Zeiterkennung. Als Regel gilt:

$$AnzahlZeitTests = \frac{Durchlaufzeit}{AuflösungUhr * 10}$$

Dadurch ist die Zeit zwischen zwei Zeittests immer ca. 10 mal länger als die Auflösung der internen Uhr. Dies bedeutet, dass der Zeittest 90% der Sprünge die einen falschen zeitlichen Ablauf hervorrufen entdeckt.

Eine geringere Anzahl von Zeittests führt zu einer höheren Aufdeckungsrate für einzelne Fehlerhafte Sprünge. Allerdings steigt gleichzeitig die Wahrscheinlichkeit, dass Doppelsprünge ihre Fehlerzeit gegenseitig ausgleichen.

10.3.2.2 Algorithmus

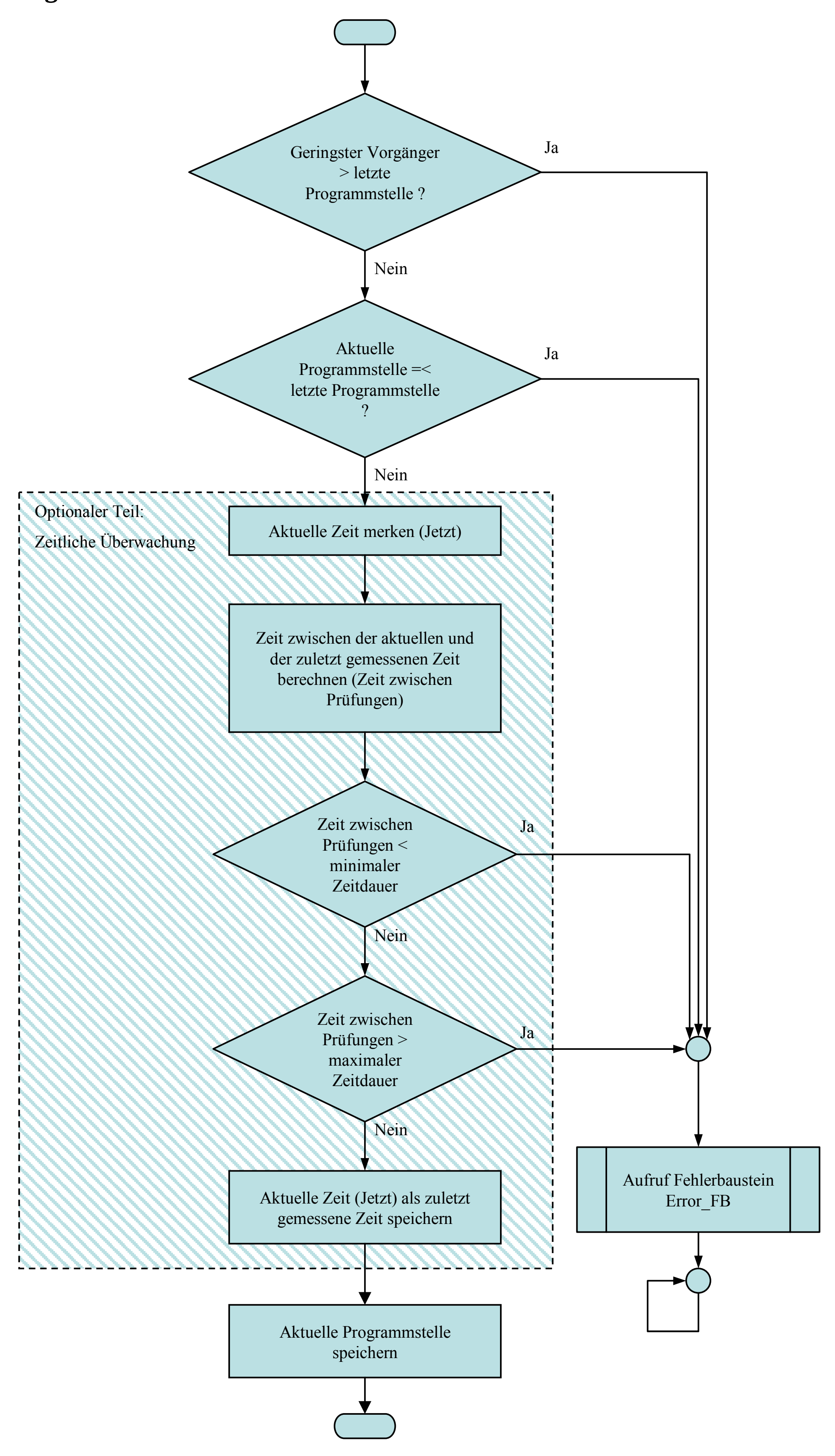


Abbildung 47: Flussdiagramm Ablaufüberwachung

Der Hauptbaustein der Ablaufüberwachung aus Abbildung 47 prüft zuerst die aktuelle Position. Dazu vergleicht er die Nummer des zuletzt aufgerufenen Bausteins mit der Variablen *GeringsterVorgaenger*. Hiernach vergleicht er die eigene *Programmstelle* mit der *Programmstelle* des zuletzt aufgerufenen Bausteins, welche in der globalen Variablen *LetzteProgrammstelle* abgespeichert ist. Stellt er in diesen Prüfungen einen falschen Zustand fest, wird die Fehlerroutine aufgerufen.

In den Funktionsbausteinen, die eine Zeitüberwachung enthalten, wird danach die Zeit geprüft, die nach dem letzten Aufruf einer Prüfroutine vergangen ist. Dazu berechnet der Baustein die Differenz aus der aktuellen Laufzeit der SPS und der davor gemessenen Laufzeit, die in der globalen Variablen *LetzterPruefzeitpunkt* gespeichert ist. Durch die Differenzbildung und die Verwendung einer Variablenart, die nur positive Zahlen speichert (Double Word) gibt es keine Probleme mit Variablenüberläufen am Ende des Wertigkeitsbereichs. Liegt die Zeitdifferenz im vorgegebenen Zeitfester, wird die aktuelle Systemzeit in die globale Variable *LetzterPruefzeitpunkt* gespeichert. Funktionsbausteine ohne Zeitüberwachung überspringen diesen Teil der Prüfung. Beide Funktionsbausteine – mit und ohne Zeitüberwachung – speichern zum Schluss die aktuelle *Programmstelle* in die globale Variable *LetzteProgrammstelle*.

10.3.2.3 Diagnosedeckungsgrad

Dieser Test deckt im Gegensatz zum Watchdog nur Fehler im Programmzähler (siehe Kapitel 9.2.2) auf.

Der Test besteht dabei aus der zeitlichen Überwachung durch die interne Uhr der SPS und aus dem Abprüfen des Programmablaufs auf die korrekte Reihenfolge.

Zeitliche Überwachung:

Die Zeitliche Überwachung des Tests deckt zufällig auftretende Fehler nur in Bits höherer Wertigkeit (z. B. 128, 256, 512, …) des Programmzählers auf, da zufällige Fehler in Bits niederer Wertigkeit (z. B. 1, 2, 4, 8) die Zeit bei Sprüngen nicht stark genug beeinflussen. Davon ausgehend, dass ein Befehl 50µs dauert und ein Sprung ab 1,5ms entdeckt wird, müssen 30 Befehle übersprungen werden, um eine Zeitdifferenz von 1,5ms im Programmablauf auszulösen und diesen Fehler damit erkennbar zu machen. Die unteren 5 Bit des Programmzählers lösen deshalb aufgrund ihrer Wertigkeit (1, 2, 4, 8, 16) keinen Fehler aus, da sie den Befehlszähler beim Auftritt eines Fehlers maximal um 16 Befehle verfälschen können.

Von einer Programmzählerbreite von 16 Bit ausgehend, bedeutet dies eine ca. 66%ige (11 aus 16 Bit) Aufdeckung von zufälligen Fehlern, bei 32 Bit ca. 85%ige (27 aus 32 Bit). Diese Art der Berechnung für zufällige Fehler kann in der Literaturquelle [4] nachgelesen werden.

„Stuck at" Fehler im Programmzähler verkürzen die Ablaufzeit dagegen um die Hälfte. Damit werden sie mit großer Wahrscheinlichkeit (99%) entdeckt, falls das Programm eine genügende Anzahl an Befehlen pro Zyklus enthält, um die Zeitüberwachung sinnvoll einzusetzen. Dies bedeutet, dass die

Zykluszeit des Programms mindestens 10-mal länger sein muss, als die mit der konkreten SPS kürzest messbare Zeitspanne. (siehe Kapitel 10.3.2.1).

Abprüfen des Programmablaufs:

Die Wahrscheinlichkeit der Aufdeckung von zufällig auftretenden Fehlern durch die Ablaufüberwachung ist abhängig von dem Verhältnis der Anzahl der eingesetzten Tests zur Gesamtlänge des Programms. Wird alle hundert Befehle eine Ablaufkontrolle durchgeführt, so beträgt die Wahrscheinlichkeit für das Finden eines zufälligen Fehlers im fünften Bit, das 16 Befehle überspringt 16%, im Vierten 8%, usw. Dies gilt allerdings nur für den Fall, das der Zufällige Sprung in einem Codesegment auftritt, das selbst keine anderen Sprünge in Unterprogramme oder ähnliches vorsieht. In einem stark verzweigten Programm ist die Aufdeckungsrate entsprechend höher, da hier das Überspringen von Sprungbefehlen zu wesentlich größeren Verschiebungen im Programmablauf führt.

Gesamt DC:

Die Gesamteffizienz des Tests kann für dauerhafte Fehler im Programmzähler mit 99% (s. o.) und für zufällig auftretende Fehler mit 66% bzw. 85% (s. o.) angenommen werden. Ein Wert von 90% für zufällige Fehler kann mit viel Aufwand und durch häufige Tests – bei allerdings hohem Performanceverlust – durch häufiges Abprüfen des Programmablaufs erreicht werden.

10.4 Fehlererkennung durch Prozessortest

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	keine
Namen der Nebenfunktionen	JMP_TEST ACC_TEST LO_AND_TEST LO_ANDN_TEST LO_NOT_TEST LO_OR_TEST LO_ORN_TEST LO_XOR_TEST LO_XORN_TEST AR_ADD_TEST AR_DIV_TEST AR_MOD_TEST AR_MUL_TEST AR_SUB_TEST CO_EQ_TEST CO_GE_TEST CO_GT_TEST CO_LE_TEST CO_LT_TEST CO_NE_TEST LOAD_STORE_TEST
Kurzbeschreibung	Testbausteine für einzelne Funktionen des Prozessors.
Eingänge	
keine	
Ausgänge	
keine	
Genutzte globale Variablen	
keine	

10.4.1 Voraussetzung

Die Vorraussetzung für einen Großteil dieser Tests ist das Funktionieren von Sprungbefehlen und dem Programmzähler. Der Programmzähler wurde bereits im Kapitel 10.3.2 getestet. Ein nicht funktionierender Sprungbefehl kann von einer einzelnen SPS nicht erkannt werden, so dass hier vorausgesetzt werden muss, dass er funktioniert. Obwohl ein Fehler im Sprungbefehl im Programm große Auswirkungen hat, kann er nur durch die in Kapitel 10.8 beschriebenen Tests mit in Verbindung mit einer zweiten SPS erkannt werden.

10.4.2 Erläuterungen

Der Prozessortest ist auf verschiedene Funktionsbausteine aufgeteilt, die jeweils eine bestimmte Funktion des Prozessors testen. Hierdurch ist es dem Programmierer möglich, immer nur die Tests für die Funktionen einzubinden, die er in seinem Programm nutzt. Dadurch kann die Rechnerzeit, die den Testbausteinen vom Programmierer zur Verfügung gestellt wird, optimaler genutzt werden.

10.4.2.1 T2b) Test der bedingten Sprünge

Um die bedingten Sprünge zu testen, die von den Werten WAHR (1) oder FALSCH (0) ausgelöst werden, wird zunächst die Bedingung WAHR in das AE geladen. Danach wird getestet, ob nur der dem Wert WAHR zugeordnete bedingte Sprungbefehl ausgelöst wird und nicht der dem Wert FALSCH zugeordnete. Das gleiche Verfahren wird danach mit umgekehrten Werten (FALSCH und WAHR werden getauscht) erneut durchgeführt. In beiden Fällen wird bei falscher Ausführung eines bedingten Sprungs der Fehlerbaustein ERROR_FB aufgerufen.

10.4.2.2 T3) Test des Akkumulators

Durch das Fehlen eines Statusregisters in der SPS und hierbei insbesondere durch das Fehlen eines Bits, das einen Überlauf im Akkumulator während der letzten Rechenoperation anzeigt, kann der in Kapitel 9.2.3 beschriebene allgemeine Test für Microprozessoren nicht übernommen werden.

Eine SPS verfügt, wie in Kapitel 8.3.1 erwähnt, über keinen Akkumulator. Statt dessen wird dieser durch das AE simuliert. Es ist trotzdem wichtig, die vom Prozessor als Akkumulator benutzte Speicherstelle (AE) wie einen solchen zu testen. Ein Speichertest arbeitet nicht so gründlich wie ein Akkumulatortest und würde Fehler, die in diesem auftauchen, nicht bemerken.

Der Akkumulator wird dadurch getestet, dass eine Eins in das höchstwertige Bit geladen wird. Danach wird diese Eins schrittweise bis in das niedrigstwertige Bit „geshiftet“, das Ergebnis nach Durchführung aller shift-Operationen durch den Vergleich mit dem Erwartungswert geprüft. Danach wird dieser Vorgang in entgegen gesetzter Richtung wiederholt. Beim Vorliegen eines unerwarteten Ergebnisses bricht der Test ab und ruft den Fehlerbaustein ERROR_FB auf.

Die Idee hinter diesem Test liegt darin, dass ein „Stuck at“ Fehler also eine nicht funktionierende Speicherzelle entweder die durch das shiften wandernde Eins verschluckt, oder sie vervielfältigt. (siehe Abbildung 48)

Start	1	0	0	0	0	1	0	0
1 Shift nach rechts	0	1	0	0	0	1	1	0
2 Shift nach rechts	0	0	0	0	0	1	1	1
3 Shift nach rechts	0	0	0	0	0	1	1	1
4 Shift nach rechts	0	0	0	0	0	1	1	1
5 Shift nach rechts	0	0	0	0	0	1	1	1
6 Shift nach rechts	0	0	0	0	0	1	1	1
7 Shift nach rechts	0	0	0	0	0	1	1	1

Korrekt geshiftetes Bit

Beim Shiften verändertes Bit durch „Stuck at“ Fehler

Abbildung 48: „Stuck at 0“ und „Stuck at 1“ Fehler (rot) während einem Bitshift von vorne nach hinten

Nach nur einem Durchgang in eine Richtung existiert nur noch eine Fehlermöglichkeit an der letzten Stelle. Hängt hier die Eins, ist das Ergebnis wie erwartet Eins. Aus diesem Grund wird die Prüfung jeweils einmal von beiden Seiten aus durchgeführt.

10.4.2.3 T4a) Test der Logischen Operatoren

Logische Operatoren werden über ihre jeweilige Wahrheitstabelle geprüft. (siehe als Beispiel Abbildung 49)

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

Abbildung 49: Wahrheitstabelle für den Operator &

Das heißt, es wird nach der Tabelle in Abbildung 49 geprüft, ob Null & Null = Null ergibt, Null & Eins = Null und Eins & Eins = Eins.

10.4.2.4 T4b) Test der Arithmetischen Operatoren

Zum Test der arithmetischen Operatoren werden zwei konstanten Zahlen verwendet, und das durch den Operator errechnete Ergebnis mit einem im Speicher hinterlegten Erwartungswert verglichen. Die benutzten Zahlen sind dabei in ihrer binären Form möglichst abwechslungsreich. Das bedeutet, dass

zum Beispiel die dezimale Zahl 170 verwendet wird, da ihre binäre Form 1010 1010 ist und sich damit Einsen und Nullen ständig abwechseln.

10.4.2.5 T4c) Test der Komparatoren

Bei den Komparatoren werden alle möglichen Zustände jeweils einmal geprüft. Das bedeutet ein „kleiner gleich“ wird zweimal mit unterschiedlichen Zahlen getestet, die bei der korrekten Durchführung des Vergleichs einmal WAHR und einmal FALSCH ergeben müssen, sowie einmal mit identischen Zahlen die in diesem Fall als Ergebnis WAHR ergeben müssen. Das vom Komparator ausgegebene Ergebnis wird jeweils mit dem hierfür hinterlegten Erwartungswert verglichen.

10.4.2.6 T5) Test des Ladens und Speicherns von Daten

In diesem Test wird eine Konstante aus dem Programmspeicher in das AE geladen und danach als Variable abgespeichert. Der gespeicherte Wert wird hiernach erneut in das AE geladen und das Ergebnis mit der ursprünglichen Konstante verglichen.

Dies wird je einmal für eine 8 Bit Zahl, eine 16 Bit Zahl und eine 32 Bit Zahl durchgeführt, da dies die von einer SPS verwendeten Größen für Variablen sind.

10.4.3 Diagnosedeckungsgrad

Ein solcher Prozessorselbsttest, exklusive des AEs, ist vergleichbar mit dem in Tabelle A.4 der DIN EN 61508-2 [17] aufgeführten „Selbsttest per Software mit begrenzter Anzahl von Mustern“. Dieser hat die Einstufung „niedrig“ für den Diagnosedeckungsgrad, und liegt damit bei 60%.

Zusätzlich wird das AE wie in Kapitel 10.4.2.2 beschrieben mit einem „Walking Bit“ Test getestet, der in der gleichen Tabelle einen Diagnosedeckungsgrad von 90% erhält. Dies hat jedoch keine Auswirkungen auf den DC für den Prozessortest. Der Test des AEs ist zwar eng mit der Funktionalität des Prozessors verknüpft und aus diesem Grund an dieser Stelle aufgeführt, das AE selbst ist aber wie in Kapitel 6.2 beschrieben Teil des RAM-Speichers der SPS.

10.5 Fehlererkennung durch Speichertest des EEPROM

Als Speicher stehen der SPS wie in Kapitel 6.2.2 beschrieben ein EEPROM und ein RAM zur Verfügung. Von der Anwenderebene aus ist ein Zugriff aus dem Programm der SPS auf das EEPROM nicht möglich, so dass in diesem Kapitel nur der vom Hersteller integrierte Selbsttest des EEPROM beschrieben und bewertet wird.

10.5.1 T7-ROM) CRC Test des EEPROM

10.5.1.1 Erläuterung

Eine SPS prüft beim Überspielen des Programms bzw. beim Nachladen von Programmteilen vom EEPROM in das RAM den Speicherinhalt per „CRC Prüfung“. Dabei wird laut DIN EN 61508 [17] auf den Inhalt eines Speicherblocks des EEPROMs mittels eines „Generatorpolynoms“ (einem vom Hersteller festzulegenden Wert) eine laufende Polynomdivision (in diesem Fall äquivalent zur Ausrechnung des Restes bei einer Division) durchgeführt. Der Wert des Restes dieser Division wird

als Prüfsumme gespeichert. Beim Zugriff auf den Speicherbereich wird dieser Rest (CRC Summe) erneut gebildet und mit der vorhandenen verglichen.

10.5.1.2 Diagnosedeckungsgrad

Die Effizienz dieses Tests wird in der DIN EN 61508 Teil 2 Tabelle A.5 [17], in Abhängigkeit von der Länge der Prüfsumme, mit 60% (bei 8 Bit) und 90% (bei 16 Bit) angegeben.

10.6 Fehlererkennung durch Speichertest des RAM

Wie in Kapitel 10.5 schon erwähnt steht der SPS neben dem EEPROM ein RAM zur Verfügung. Auf große Teile des RAMs kann vom Anwender direkt oder indirekt zugegriffen werden, wodurch hier eigene Tests durch den Anwender implementiert werden können.

Die Aufteilung des RAM-Speichers wurde bereits in 6.3.2 beschrieben und ist an dieser Stelle in Abbildung 50 noch einmal dargestellt. Aufgrund der Einschränkung durch das Betriebssystem einer SPS kann der Anwender auf den Programmspeicher nicht zugreifen. Fehler, die in diesem Teil auftauchen, können deshalb nur indirekt mittels der Ablaufkontrolle aus Kapitel 10.3 entdeckt werden.

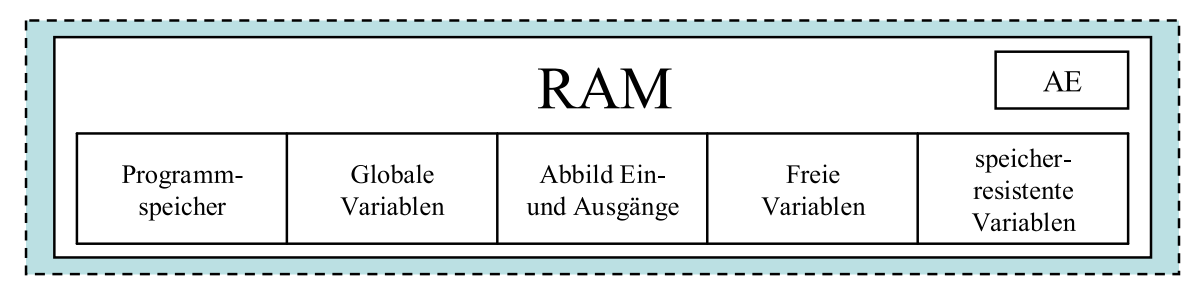


Abbildung 50: Ausschnitt aus Abbildung 33: Aufteilung des RAM-Speichers

Der Bereich der globalen Variablen des RAMs kann vom Anwender direkt adressiert und damit komplett angesprochen und somit auch getestet werden. Das gleiche gilt für den Bereich, in dem das Abbild der Ein- und Ausgänge gespeichert ist.

Im Bereich der freien Variablen und der speicherresistenten Variablen können nur die Speicherstellen angesprochen und damit getestet werden, die auch einer Variablen zugeordnet sind.

Der dem AE zugeordnete Speicherbereich, der ebenfalls im RAM untergebracht ist wird mittels des bereits im Kapitel 10.4.2.2 beschriebenen Tests des Akkumulators geprüft.

10.6.1 T6-RAM) Test der benutzten Variablen

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	VAR_TEST_RESET
Namen der Nebenfunktionen	VAR_TEST_BOOL VAR_TEST_BYTE VAR_TEST_DINT VAR_TEST_DWORD VAR_TEST_GLOBALS VAR_TEST_INT VAR_TEST_RESET VAR_TEST_SINT VAR_TEST_UDINT VAR_TEST_UINT VAR_TEST_USINT VAR_TEST_WORD
Kurzbeschreibung	Test des Speicherorts von dynamisch zugewiesenen Variablen
Eingänge	
Anzahl_Durchläufe_bis _Gesamttest	Die Anzahl der Zyklen, bis alle angegebenen Variablen einmal getestet wurden.
Ausgänge	
keine	
Genutzte globale Variablen	
Aktueller_Var_Test	Speichert die Nummer des Aktuellen Testbausteins.
Start_Var_Test	Speichert die Nummer des Testbausteins, ab dem der eigentliche Test beginnt.
Ende_Var_Test	Speichert die Nummer des letzten zu testenden Testbausteins.
Begonnen_Var_Test	Speichert ob der erste Test des Zyklus aufgerufen wurde.
Beendet_Var_Test	Speichert ob der letzte Test des Zyklus aufgerufen wurde.

*Hilfs_Variable_**	Jeder Variablentyp (*) der getestet wird benötigt eine Hilfsvariable zum sichern der eigenen Daten. Zum Beispiel: *Hilfs_Variable_DWORD*

10.6.1.1 Erläuterung

Dieser Test überprüft die Speicherstellen der freien Variablen.

Durch die oben erwähnte Einschränkung für den Bereich der freien Variablen können grundsätzlich nur die Speicherstellen getestet werden, die mit Variablen belegt sind. Zusätzlich ist zu beachten, dass diese Variablen nur von dem POE (siehe Kapitel 8.2) aus getestet werden können, in dem Sie deklariert wurden. Durch die in Kapitel 8.4 beschriebene Übergabe der Variablen per Referenz an aufgerufene Funktionsbausteine ist ein Test des Speicherbereichs dieser Variablen auch von diesen aus möglich.

Dieser Test hat mehrere Nachteile:

- Korrelationen zwischen Speicherstellen sind nicht testbar, da der Programmierer aufgrund des zufälligen Abspeicherns dieser Daten innerhalb des Gesamtbereichs nicht weiß, welche Variable an welcher Stelle des Speicherbereichs der freien Variablen abgespeichert wird. Das bedeutet dieser Test kann keine Übertragungsfehler finden.
- Der Programmieraufwand ist durch die beschriebenen eingeschränkten Zugriffsrechte hoch, da die verschiedenen Testbausteine aus jeder einzelnen POE heraus für die hier deklarierten Variabeln aufgerufen werden müssen.
- Variablen die von Funktionen benutzt werden können gar nicht getestet werden, da diese keine Funktionsbausteine aufrufen dürfen, die für den Test aber notwendig sind.

10.6.1.2 Algorithmus

Der Gesamttest besteht aus zwei Teilen:

- Teil Eins: Gesamtsteuerung des Testablaufs – Initialisierungs- und Reset-Routine
- Teil Zwei: die verschiedenen Variablen-Tests

Diese Zweiteilung ist nötig, damit in einem Programmzyklus nicht immer alle Variablen getestet werden müssen.

Ist das Programm nur klein bzw. ist die Anzahl der benutzten Variablen gering, so können die eigentlichen Variablentests auch ohne vorgeschaltete Gesamtsteuerung umgesetzt werden.

Der erste Teil des Gesamttests, die Initialisierungs- und Reset-Routine, wird in Abbildung 51 grafisch dargestellt und im folgenden Text erläutert.

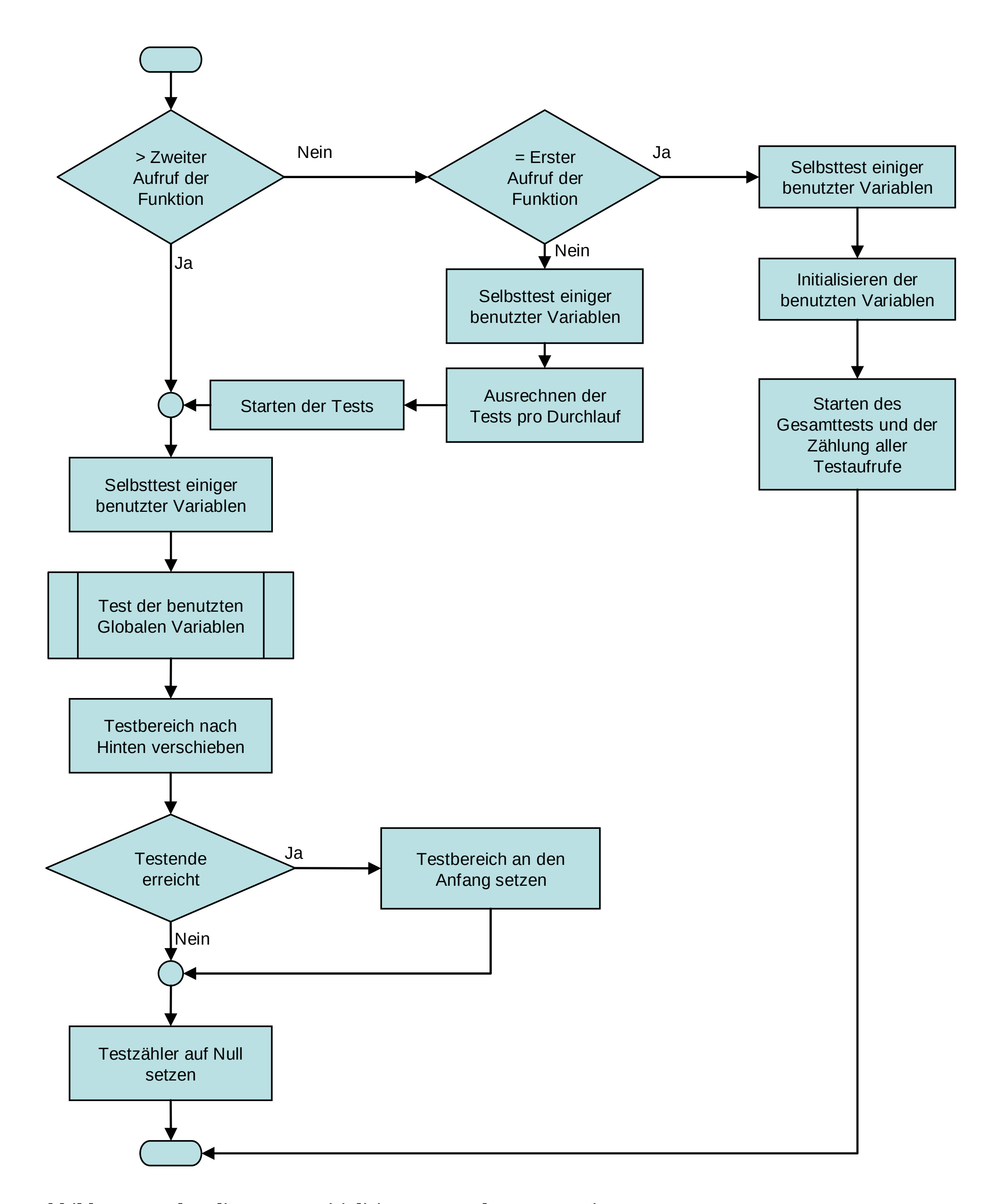


Abbildung 51: Flussdiagramm Initialisierungs- und Reset-Routine

Teil Eins des Gesamttests beinhaltet drei verschiedene Funktionen, abhängig von der Anzahl der bisherigen Aufrufe dieses Testteils (1, 2, >2). Deshalb wird zu Beginn des Teil Eins geprüft, zum wievielten Mal er aufgerufen wurde. Bei den ersten beiden Aufrufen wird die Initialisierung des Gesamttests durchgeführt. In jedem weiteren Durchgang wird geprüft ob ein Reset nötig ist, da die Speicherprüfung einen kompletten Speicherdurchlauf beendet hat, und dieser bei Bedarf durchgeführt.

Die Initialisierung im ersten Aufruf des Teil Eins führt zum Setzen der vom Gesamttest zur internen Kommunikation benötigten globalen Variablen. Diese globalen Variablen sind zur Steuerung der Testbausteine in Teil Zwei nötig. Hier werden sie so gesetzt, dass alle freien Variablen, die mit Testroutinen des Teil Zwei verknüpft sind, einmal überprüft werden.

Im zweiten Aufruf wird aus der Anzahl der Testroutinen die im ersten Zyklus gezählt wurden und der vom Programmierer vorgegebenen Anzahl der Zyklen, die ein Gesamttest aller Variablen benötigen darf, die Anzahl der benötigten Variablen-Tests pro Zyklus berechnet. Damit wird in jedem Zyklus nur eine Teilmenge aller zu prüfenden Variablen getestet. So kann die benötigte Testdauer auf mehrere Zyklen optimal verteilt werden.

Ebenfalls im zweiten Durchgang, sowie auch in jedem weiteren Durchgang, wird die Testroutine der von diesem Test genutzten globalen Variablen aufgerufen und der Testbereich auf die nächste Teilmenge der freien Variablen verschoben. Sind alle Variablen getestet wird im nächsten Zyklus wieder mit dem ersten Teilbereich begonnen. Die ebenfalls notwendige Testroutine der benutzten globalen Variablen (s. u.) wird in diesem Ablauf in den letzten Zyklus mit eingebunden. Anzumerken ist, dass sie in jedem Zyklus aufgerufen wird, ein Aufruf einer Testroutine aber nicht unbedingt zu deren Durchführung führt (s. u.).

Der Test der globalen Variablen ist ähnlich dem Test jeder anderen Variablen und wird darum hier nicht dargestellt.

Teil Zwei des Gesamttests beinhaltet die eigentlichen Variablentests, deren Ablauf in Abbildung 52 dargestellt und im nachfolgenden Text erläutert ist.

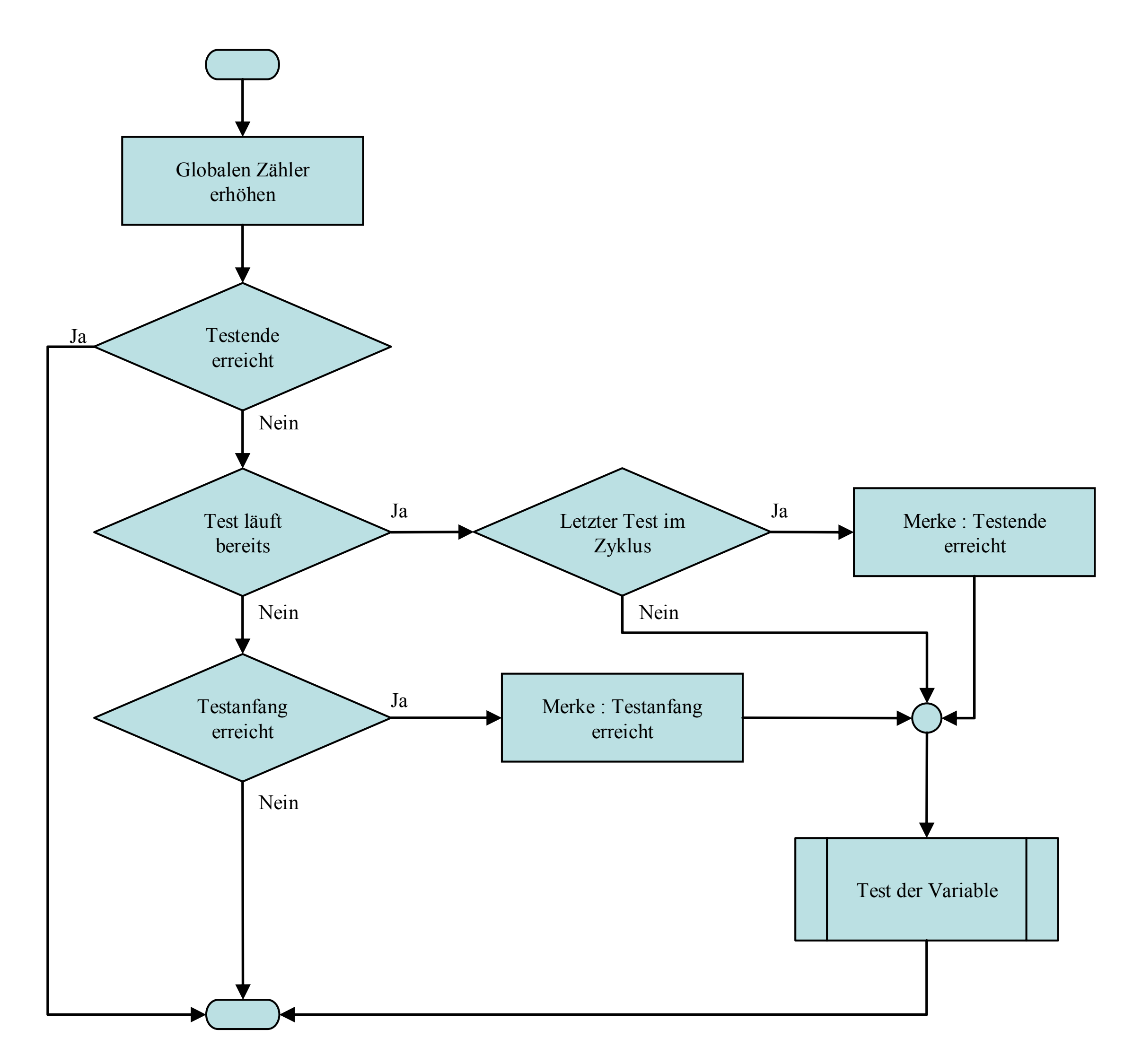


Abbildung 52: Flussdiagramm Aufruf eines Variablentests

Da die DIN EN 61131-3 eine Strenge Trennung von Variablentypen vorschreibt, wurde für jeden Variablentyp ein eigener Test programmiert. Diese einzelnen Tests gleichen sich allerdings in der Funktion und werden darum nur einmal erläutert.

Diese Testroutinen werden wie in Kapitel 10.6.1.1 beschrieben aus den POEs heraus aufgerufen, in denen die zu testende Variable deklariert wurde. Diese wird dem Testbaustein per Referenz auf ihren Speicherort übergeben.

In der Testroutine wird vor dem eigentlichen Test erst eine Überprüfung durchgeführt, ob diese Variable im aktuellen Zyklus getestet werden soll. (zur Verteilung der Variablentests auf verschiedene Zyklen siehe oben)

Als erstes wird überprüft, ob der Testlauf des aktuellen Zyklus schon beendet ist, da dann keine weiteren Funktionen mehr durchgeführt werden müssen. Dies ist eine einfache boolsche Überprüfung einer globalen Variablen, die wenig Zeit in Anspruch nimmt.

Als nächstes wird geprüft ob der Testlauf im aktuellen Zyklus bereits begonnen hat, welches ebenfalls nur die Überprüfung einer boolschen Variablen bedeutet. Hat der Testlauf noch nicht begonnen, so

wird geprüft ob er mit diesem Testaufruf beginnt. Läuft der Testablauf bereits, so wird geprüft ob dies die letzte zu überprüfende Variable des Zyklus ist. Das jeweilige Ergebnis wird global gespeichert und bietet wiederum die Grundlage der boolschen Überprüfungen des nächsten Testaufrufs.

Der nächste Teil der Routine ist der eigentliche Variablentest, der hier am Beispiel einer „ein Byte Variablen“ beschrieben wird.

Im ersten Schritt wird der Inhalt der Zelle gesichert. Dies geschieht in eine globale Speicherzelle, die im ersten Teil des Tests bereits auf ihre Funktionalität geprüft wurde.

Die nächsten vier Schritte, die in Abbildung 53 bis Abbildung 56 dargestellt sind, beinhalten je ein Beschreiben der Prüfzelle mit einem Prüfmuster und das anschließende Testen, ob sich das Prüfmuster aus der Zelle wieder erfolgreich auslesen lässt.

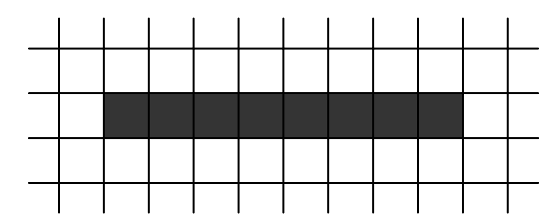


Abbildung 53: Schritt 2: Die Prüfzelle wird mit Einsen gefüllt.

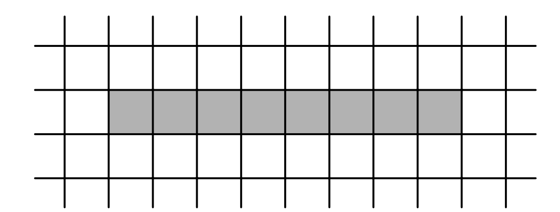


Abbildung 54: Schritt 3: Die Prüfzelle wird mit Nullen gefüllt.

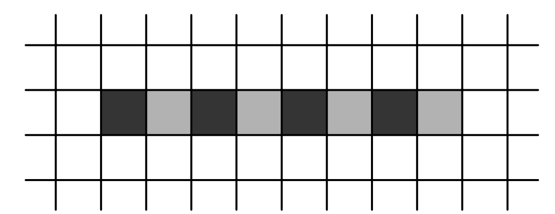


Abbildung 55: Schritt 4: Die Prüfzelle wird abwechselnd mit Null und Eins beschrieben.

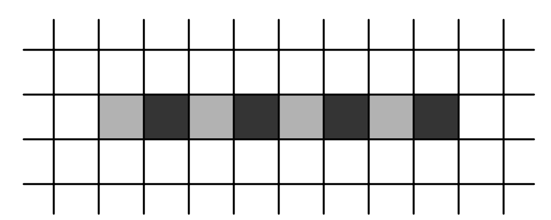


Abbildung 56: Schritt 5: Das Muster aus Schritt 4 wird komplementiert.

Kommt es in einem dieser vier Schritte beim Testen zu einem Fehler, so wird sofort der Fehlerbaustein ERROR_FB aufgerufen und die SPS damit abgeschaltet.

Bei erfolgreichem Test ist Schritt 6 das Rückschreiben des ursprünglichen Zelleninhalts.

10.6.1.3 Diagnosedeckungsgrad

Die Effizienz dieses Testverfahrens ist nur gering. Fehler im Speicher, die durch Übertrag entstehen (siehe Kapitel 9.1) können mit dieser Art von Test wie im Kapitel 10.6.1.1 beschrieben nicht gefunden werden. Aus diesem Grund wird dieser Art von Test von der Norm DIN EN 61508 nur ein Diagnosedeckungsgrad von 60% zugesprochen.

10.6.2 T7a-RAM) Test des Bereichs der globalen Variablen mittels eines Arrays

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	VAR_TEST_GLOBAL
Namen der Nebenfunktionen	VAR_TEST_GLOBAL_CHKSUM VAR_TEST_GLOBAL_CHKSUM_SAVEVAR VAR_TEST_GLOBAL_SNS VAR_TEST_GLOBAL_TEST
Kurzbeschreibung	Test des Bereichs der globalen Variablen des RAM-Speichers.
Eingänge	
Feldbreite	Breite des vorbelegten Bereichs der globalen Variablen
AnzahlTestsproDurchlauf	Anzahl der Variablen die pro Zyklus getestet werden
Testbreite	Gesamtbreite der Signatur für die Suche nach Übertragungsfehlern
Ausgänge	
finished	Gibt zurück, ob ein kompletter Durchlauf abgeschlossen wurde.
Genutzte globale Variablen	
SaveVar	Speichert den Inhalt der zu testenden Zelle.
SignatureVar	Speichert die Signatur zum späteren Vergleich.
TestFeld	Ein Array über den gesamten Bereich der globalen Variablen, der es erlaubt jede Speicherstelle im Bereich der globalen Variablen anzusprechen.

10.6.2.1 Erläuterung

Die zweite Möglichkeit des Speichertests bedient sich der in Kapitel 8.4 beschriebenen Möglichkeit von Doppelbelegung im Bereich der globalen Variablen.

Wird dieser Test eingesetzt, so müssen alle im eigentlichen Programm genutzten sicherheitsrelevanten Variablen vom Programmierer im Bereich der globalen Variablen angelegt werden. Über den gleichen Bereich, und einen darüber hinaus gehenden Sicherheitsbereich, wurde für den in dieser Diplomarbeit programmierten Test ein Array bestehend aus Variablen der Variablenart „Double Words“ angelegt. Dieser Array kann maximal den gesamten Bereich der globalen Variablen umfassen. Dies ist die größte Variablenart (32 bit) die nach DIN EN 61131-3 verwendet werden kann. Hierbei wurde davon ausgegangen, dass der Prozessor der SPS ein 32 bit Prozessor ist und somit solche Variablen in einem Schritt verarbeiten kann. Ist dies nicht der Fall, so muss auf kleinere Variablenarten (16 bit, 8 bit) zurückgegriffen werden. Weiterhin belegt dieser Test zwei Speicherstellen gleicher Art vor dem angelegten Array, die er selbst für das Sichern der geprüften Speicherstelle, sowie für das Sichern der Signatur benötigt.

Ein Array ist eine Menge von Variablen derselben Variablenart, die in einer bestimmten Struktur angeordnet sind. In der Programmierung werden in der Regel eindimensionale (Variablen in Reihe angeordnet), zweidimensionale (Variablen im Feld angeordnet) und dreidimensionale (Variablen im Raum angeordnet) Arrays genutzt. Der Array der für diesen Test angelegt werden soll muss eindimensional sein.

Durch diesen Array ist es möglich die Speicherprüfung in einer Schleife laufen zu lassen, da die Nummer des Anzusprechenden Speicherbereichs als Variable an den Array übergeben werden kann und somit vom Programm veränderbar ist. Im Prinzip wurde dadurch ein Pointer simuliert.

Der Nachteil dieser Testart ist, dass der Programmierer streng darauf achten muss, welche globale Variable welchen Speicherplatz benutzt. Die Methode der festen Zuweisung ist fehleranfällig, und muss deswegen genau kontrolliert werden. (siehe Kapitel 8.4 und hier besonders Abbildung 37)

In der Programmierumgebung CoDeSys kann der Programmierer sich Doppelbelegungen anzeigen lassen und hat damit eine gute Möglichkeit das Programms auf fehlerhafte Doppelbelegungen zu überprüfen.

In diesem Test und im Test in Kapitel 10.6.3 werden Signaturen über den Speicherbereich gebildet, der auf Übertragungsfehler (siehe Abbildung 38) geprüft wird. Signaturen sind mathematische Verknüpfung von Zahlen, hier dem Inhalt der überwachten Speicherzellen, die durch diese Verknüpfung ein möglichst eineindeutiges Ergebnis bilden. Durch den Vergleich dieses berechneten Wertes mit einem festgelegten oder vorher berechneten Wert kann die Signatur überprüft werden. Die gewählte Methode in diesem Fall ist eine „XOR-Verknüpfung“. Das bedeutet jede Speicherzelle wird mit jeder anderen über die „exklusiv-oder-Funktion“ verknüpft und daraus die Signatur errechnet. (siehe Abbildung 57)

Zelle 1	1	0	1	1	0	0	1	0
Zelle 2	0	1	1	0	0	0	1	1
Zelle 3	1	1	0	0	1	0	1	0
XOR	0	0	0	1	1	0	1	1

Abbildung 57: Signaturbildung über 3 Speicherzellen

Ändert sich zwischen der Bildung zweier Signaturen der Inhalt einer ungeraden Anzahl von Speicherzellen, so wird dies durch den Vergleich der Signaturen erkannt. Eine Änderung einer graden Anzahl von Speicherzellen bleibt unerkannt. (siehe Abbildung 58) Dieser Mangel ist typisch für eine Signaturbildung mit der „XOR-Funktion“. In den beschriebenen Tests wird dies dadurch aufgefangen, dass das Testmuster von Test zu Test sich um jeweils eine Speicherstelle verschiebt. (siehe Abbildung 60) Dadurch wird in einem der folgenden Schritte eine der defekten Speicherstellen von der Signatur nicht mehr abgedeckt und aus der dann geraden Anzahl an defekten Speicherstellen wird eine ungerade, die wiederum erkannt wird.

Zelle 1	1	0	1	1	0	0	1	1
Zelle 2	0	0	1	0	1	0	1	0
Zelle 3	1	0	0	0	1	0	1	1
XOR	0	0	0	1	0	0	1	0

Geänderte Speicherzelle

Unerkannter Doppelfehler

Erkannter Einzelfehler

Erkannter Dreifachfehler

Abbildung 58: zweite Signaturbildung über 3 Speicherzellen mit Fehlern (vgl. Abbildung 57)

10.6.2.2 Algorithmus

Der Test gliedert sich in eine Hauptfunktion und vier Nebenfunktionen auf. Die Hauptfunktion ist in Abbildung 59 im Flussdiagramm dargestellt.

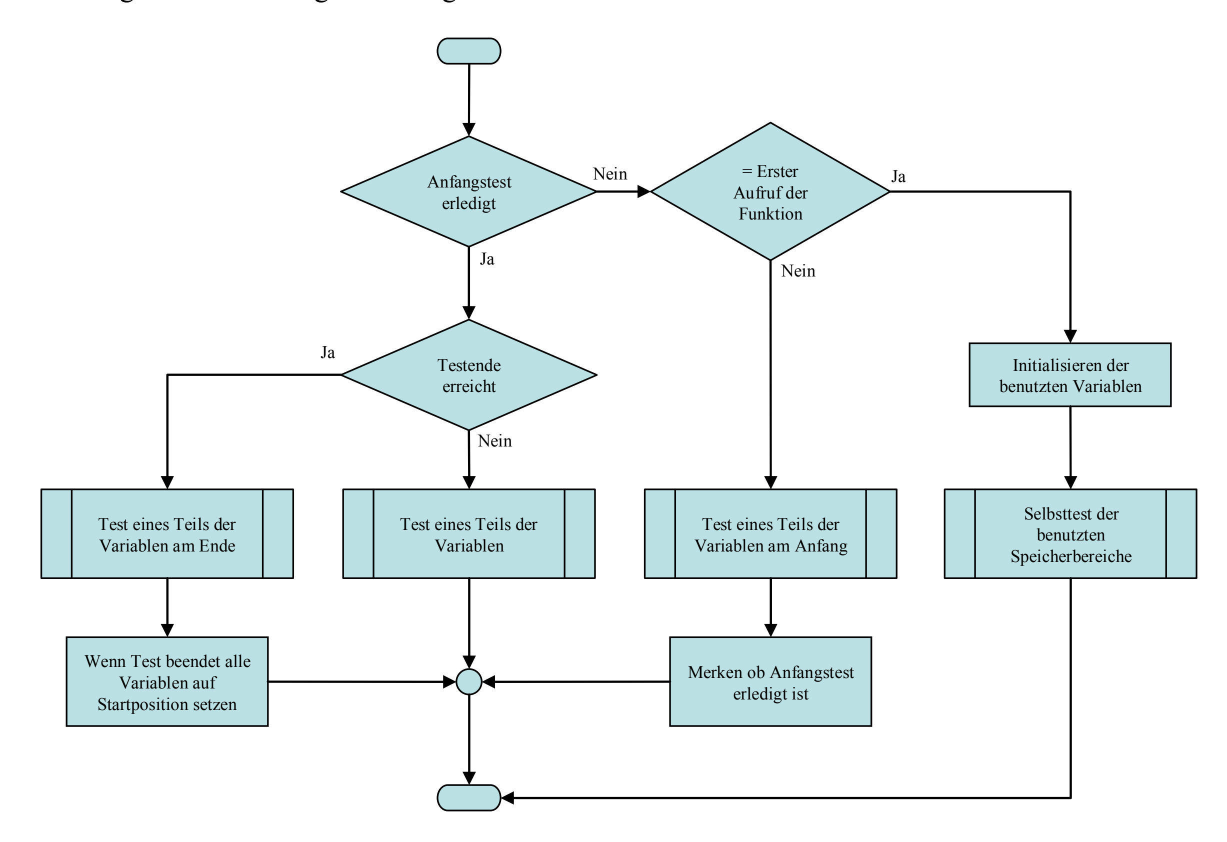
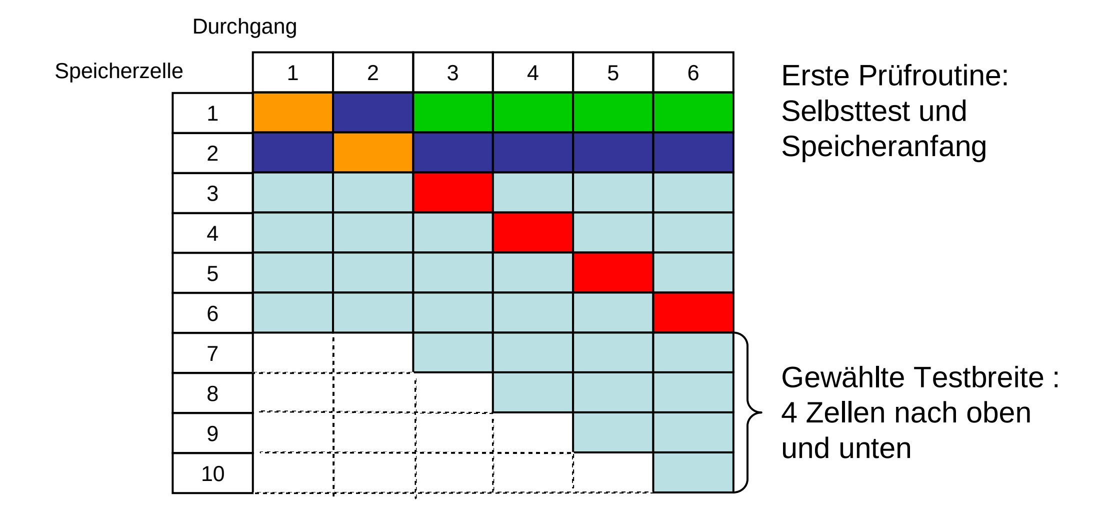


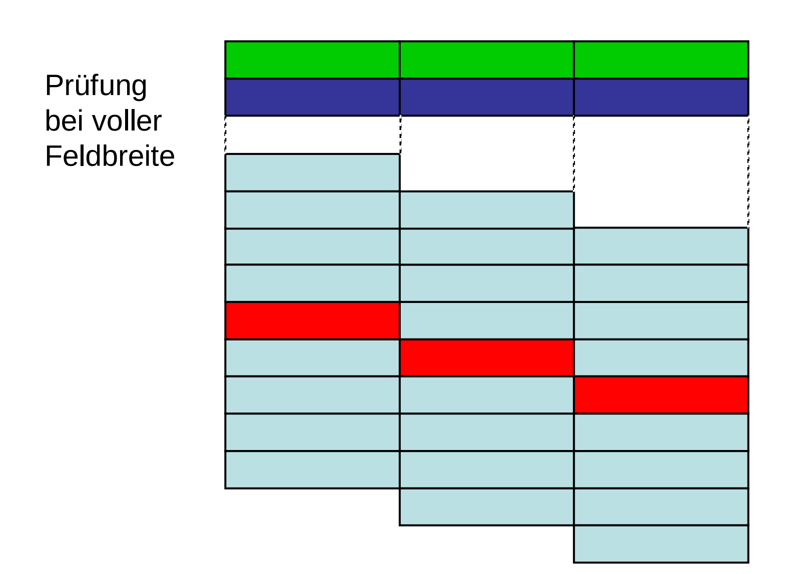
Abbildung 59: Flussdiagramm Test des Bereichs der globalen Variablen

Der Test des Bereichs der globalen Variablen, der in Abbildung 60 grafisch dargestellt ist, läuft zyklisch über den vom Array belegten Speicherbereich von der ersten belegten Speicherstelle bis zur letzten. Dabei werden im ersten Durchgang des Test-Zyklus die benutzten Speicherstellen zur Sicherung des Zelleninhalts der zur testenden Zelle und die Speicherstelle zur Speicherung der Signatur geprüft. Dies kann ohne Sicherung des Inhaltes der beiden Zellen geschehen, da hier keine Werte anderer Programmteile stehen können. Beim Überprüfen der Speicherstelle der Signatur wird die Signatur in der Speicherstelle zum Sichern des Inhalts gespeichert.

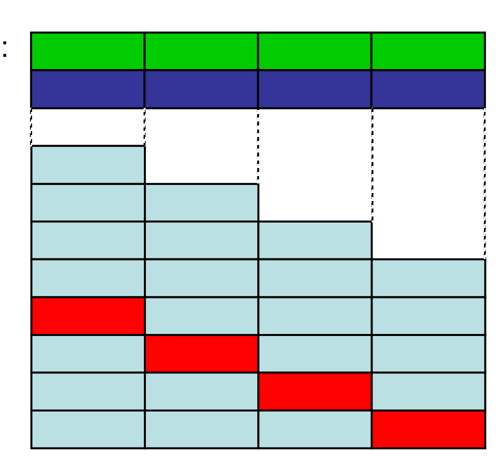
Für den Anfangsbereich und den Endbereich des Arrays muss jeweils berechnet werden, wie viele Variablen zur Bildung der Signatur oberhalb bzw. unterhalb der zu überprüfenden Speicherstelle innerhalb des Arrays liegen und somit angesprochen werden können.

Nach dem Test im Anfangsbereichs des Arrays (s. u.), in dem wie beschrieben nicht die volle Signaturbreite im vorderen Bereich der Signatur genutzt werden kann, folgt der Test des mittleren Teils des Arrays (s. u.) mit voller Signaturbreite. Im Endbereich des Arrays (s. u.) wird die Signaturbreite im hinteren Teil der Signatur reduziert und der Test für das Ende des Testbereichs aufgerufen. Nach dem Abschluss des letzten Tests werden alle vom Test genutzten Variablen zurückgesetzt, so dass der Gesamttest beim nächsten Aufruf wieder von Anfang an beginnt.





Letzte Prüfroutine:
Speicherende



Test ohne Sichern

Test mit Sichern

Signaturbildung

Speicherung der Signatur

Kopie der Testvariable

Abbildung 60: Grafische Darstellung des Verlaufs einer Prüfung im Bereich der globalen Variablen

Im Folgenden wird eine Testroutine zum Prüfen des Inhalts einer einzelnen Speicherstelle innerhalb des Arrays beschrieben, deren Flussdiagramm in Abbildung 61 dargestellt ist.

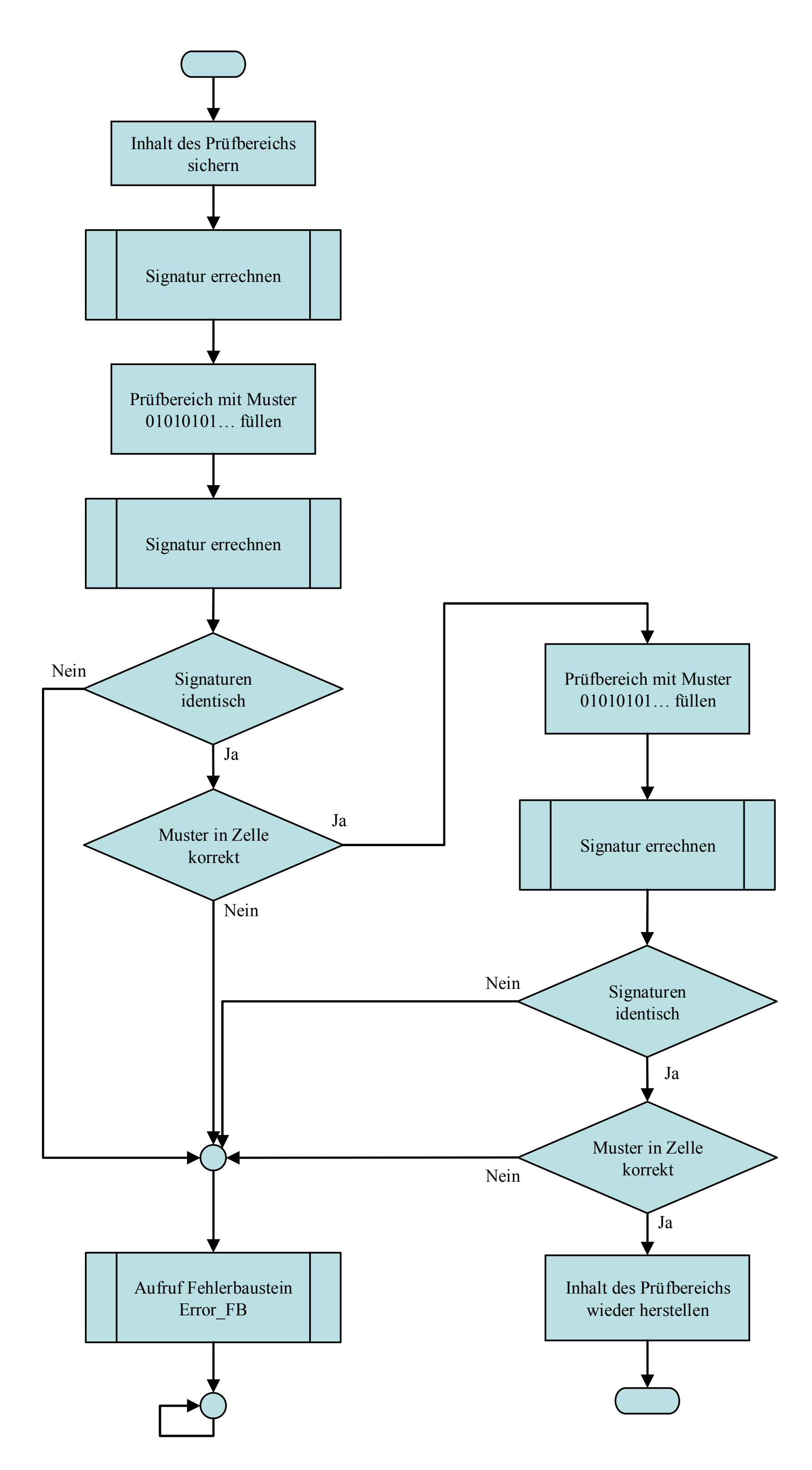


Abbildung 61: Flussdiagramm Testroutine: Prüfen des Inhalts einer Speicherstelle

Die Testroutine sichert als erstes den Inhalt der zu prüfenden Speicherstelle in die Speicherstelle der dafür angelegten globalen Variable *SaveVar*. Hierbei ist es, wie in Kapitel 8.4 beschrieben, unerheblich, welcher Datentyp eigentlich in der Zelle vorhanden ist.

Hiernach wird die Signatur über die festgelegte Anzahl an Speicherstellen, bei unverändertem Speicherinhalt der zu testenden Speicherstelle, gebildet und in der globalen Variablen *SignatureVar* gespeichert.

Der Test besteht darin, dass die Prüfzelle einmal mit dem Muster 0101 0101… und einmal mit dem komplementären Muster 1010 1010… beschrieben wird. Dazwischen wird jeweils der geschriebene Wert ausgelesen und mit dem Eingangsmuster auf Übereinstimmung überprüft. Zusätzlich wird die Signatur der umliegenden Speicherstellen erneut gebildet und mit der ursprünglichen Signatur verglichen. Dabei führt jede Abweichung von Signaturwerten oder geschriebenen Werten zu deren Originalwert zum Aufruf des Fehlerbausteins ERROR_FB.

Am Ende des Tests wird der ursprüngliche Inhalt der Zelle wiederhergestellt.

Zur Bildung der Signatur mittels eines Funktionsbaustein siehe die Ausführung in Kapitel 10.6.2.1.

10.6.2.3 Diagnosedeckungsgrad

Diese Art des Speichertests kommt dem in der DIN EN 61508-2 in Tabelle A.6 [17] angegebenen „Walk Path“ sehr nahe. Im Gegensatz zum „Walk Path“ verzichtet der umgesetzte Test aber auf eine Vorbelegung Speicherbereichs, der zur Signaturbildung dient, mit festgelegten Werten. Der „Walk Path“ ist zur Prüfung eines unbelegten Speichers konzipiert und legt deshalb die Werte in den zu prüfenden Zellen selbst an, um mögliche Fehler entdecken zu können. Der umgesetzte Test verzichtet hierauf, da der Speicher zur Testzeit genutzt wird und somit ein natürliches Muster vorhanden ist. Dies nimmt dem Test wenn überhaupt nur wenig von seiner Effektivität. Dies ist jedoch vernachlässigbar.

Der „Walk Path“ hat laut Norm eine Testgenauigkeit von 90%. Der Umgesetzte Test wird auf Grund seiner Ähnlichkeit ebenfalls mit 90% angenommen, kann allerdings nur den Bereich der globalen Variablen testen. Beachtet werden sollte auch, dass Übertragungsfehler, die durch an den Bereich der globalen Variablen angrenzende Speicherstellen entstehen, nicht erkannt werden können. Aus diesem Grund gelten die angegebenen 90% nur für den inneren Bereich des geprüften Arrays.

Als innerer Bereich wird hier der Bereich des Speichers bezeichnet, der physikalisch gesehen genügend Abstand von anderen Speicherbereichen hat, so dass Übertragsfehler ausgeschlossen werden können.

10.6.3 T7b-RAM) Test des Speichers mittels eines Zeigers (CoDeSys-spezifisch)

Dieser Test verwendet Pointer, sowie dem normalen Befehlsumfang (nach DIN EN 61131-3) hinzugefügte Operatoren, die den physikalischen Speicherort einer Variablen im RAM-Speicher ausgeben. Diese CoDeSys-spezifisch Befehle sind auf einer großen Gruppe von SPSen verschiedener Hersteller, die in der CAA zusammengeschlossen sind (siehe Kapitel 8.6), umgesetzt. Weiterhin kann der Test auch auf andere SPSen übertragen werden, die Funktionen dieser Art unterstützen. Aus diesem Grund ist der Test in die Diplomarbeit mit aufgenommen worden, auch wenn er nicht in jedem Einzellfall angewandt werden kann.

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	VAR_TEST_POINTER
Namen der Nebenfunktionen	VAR_TEST_POINTER_ASYNCHONOUSFIELD VAR_TEST_POINTER_FIELD VAR_TEST_POINTER_OWN
Kurzbeschreibung	Test des RAM-Speichers über Pointer.
Eingänge	
Anfangsstelle	Pointer auf die erste zu prüfende Stelle
Feldbreite	Breite des zu prüfenden Speichers
AnzahlTestsproDurchlauf	Anzahl der Variablen die pro Zyklus getestet werden
Testbreite	Gesamtbreite der Signatur für die Suche nach Übertragungsfehlern
Ausgänge	
finished	Gibt zurück, ob ein kompletter Durchlauf abgeschlossen wurde.
Genutzte globale Variablen	
SaveVar	Speichert den Inhalt der zu testenden Zelle.
SignatureVar	Speichert die Signatur zum späteren Vergleich.

10.6.3.1 Erläuterung

Die Programmierumgebung CoDeSys, die in Kapitel 8.6 vorgestellt wurde, erlaubt wie oben erwähnt im Gegensatz zur DIN EN 61131-3 den Einsatz von Pointern auf Variablen und das Laden der Adresse einer Variablen in den Akkumulator.

Durch diese Zusatzfunktionen ist es möglich, jeden Speicherbereich im RAM, mit Ausnahme des Programmspeichers, direkt anzusprechen.

Beim Programmieren muss insbesondere darauf geachtet werden, keinen Pointeraufruf zu programmieren, der in einen ungültigen Speicherbereich (z. B. hinter die letzte Zelle des Speichers) zeigt. Dieser würde zu unvorhersehbaren Reaktionen der SPS, vom Laden eines zufälligen Wertes bis zum Absturz der SPS, führen.

In der Soft-SPS von CoDeSys 2.3, der aktuellen Version von CoDeSys, und in den meisten mit CoDeSys programmierten SPSen liegt der fest vergebene Bereich der globalen Variablen vor dem frei

belegbaren Speicher. Dadurch kann die erste physikalische Speicherstelle gefunden werden, indem die Adresse der ersten globalen Variablen „%MB0“ ausgelesen wird. Der Quellcode hierfür lautet:

```
LD %MB0     (*Lade in den Akkumulator den Wert der Speicherstelle Erstes Byte
            Bereich der globalen Variablen*)
ADR         (*Lade in den Akkumulator die Position der geladenen
            Speicherstelle – CoDeSys spezifischer Befehl*)
```

10.6.3.2 Algorithmus

Die Testroutine dieses Tests ist im Ablauf ähnlich der Testroutine des Bereichs der globalen Variablen aus Kapitel 10.6.2. Die Unterschiede der beiden Verfahren werden nachfolgend erläutert.

Der Test des Bereichs der globalen Variablen (Kapitel 10.6.2) deckt nur diesen einen Bereich des RAM ab. Der in diesem Kapitel beschriebene Test besitzt mindestens diese Speicherabdeckung, da im Bereich der globalen Variablen über die direkte Adressierung der Anfang und das Ende des Bereichs relativ problemlos erfasst werden können. Besitzt der Programmierer weitere Kenntnisse über das Layout der einzelnen Speicherbereiche im RAM, so können von diesem Test auch darüber hinausgehende Speicherbereiche (insbesondere der Bereich der freien Variablen) erfasst werden.

Die ersten sechs Stellen des Bereichs der globalen Variablen werden von diesem Test für:

- zwei Signaturspeicher
- einen Speicher zum Sichern des Inhalts der zu prüfenden Speicherstelle
- interne Zählvariablen

verwendet. Diese Speicherstellen dürfen von diesem Test nicht über die eigentliche Testroutine geprüft werden, sondern müssen durch die testeigene Selbstprüfung überwacht werden.

Wie beim Test der globalen Variablen legt der Programmierer eine Signaturbreite fest. Diese gibt an, wie viele Speicherstellen vor und hinter der zu testeten Variablen der Signaturbildung dienen. Zusätzlich legt der Programmierer die erste zu prüfende Speicherstelle und die Länge des hinter dieser Stelle zu überprüfenden Bereichs fest.

Die verschiedenen Speicherbereiche, die mit diesem Test getestet werden können, liegen im RAM physikalisch nicht unbedingt hintereinander. Sie können durch Speicherbereiche, die nicht getestet werden können (z. B. Programmspeicher) getrennt sein. Um solche nicht zusammenhängenden Bereiche im Speicher nacheinander testen zu können, gibt der Test in einer dafür vorgesehenen Variablen an, ob er den Gesamttest des ihm jeweils zugewiesenen Speichersegments im aktuellen Zyklus abgeschlossen hat. Mit Hilfe dieser Information können mehrere Speichersegmente so verknüpft werden, dass sie, aufgeteilt auf mehrere aufeinander folgende Zyklen, der Reihe nach komplett getestet werden. Ebenfalls kann dadurch dafür gesorgt werden, dass bei jedem Einschalten der SPS ein Komplettest durchlaufen wird, der den gesamten Speicher vor dessen Nutzung im Anwenderprogramm testet.

10.6.3.3 Diagnosedeckungsgrad

Die Testgenauigkeit dieses Tests liegt, wie auch die des Tests aus Kapitel 10.6.2, bei 90%. Beide Testverfahren unterscheiden sich nicht in der Art des Testalgorithmus. Sie unterscheiden sich allerdings leicht in ihrer Programmierung (Array / Pointer) und stark in der Anwendbarkeit der Testverfahren auf die verschiedenen Speicherbereiche, da der CoDeSys-Test im Gegensatz zum Test aus Kapitel 10.6.2 fast im gesamten RAM einsetzbar ist.

Im Rahmen des praktischen Vergleichs beider Tests hat sich dieser Test auf der Soft-SPS von CoDeSys auch als um 10% schneller erwiesen.

Wegen der besseren Speicherabdeckung und der höheren Geschwindigkeit ist dieser Test dem Test aus Kapitel 10.6.2 auf jeden Fall vorzuziehen.

10.7 Fehlererkennung durch Test der Ein- und Ausgabe

10.7.1 T6-EA) Test des Speichers der Ein- und Ausgänge

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	O_TEST
Namen der Nebenfunktionen	
Kurzbeschreibung	Test des Speichers der Ein- und Ausgänge.
Eingänge	
keine	
Ausgänge	
keine	
Genutzte globale Variablen	
keine	

10.7.1.1 Erläuterung

Dieser Test läuft wie der eigentliche Variablentest innerhalb des Tests der benutzten Variablen aus Kapitel 10.6.1 ab. (siehe hierzu Abbildung 53 bis Abbildung 56)

Die Einfachheit des Algorithmus dieses Testverfahrens liegt dabei nicht in der Ansprechbarkeit dieses Variablenbereichs begründet, sondern in dessen Kürze. Dieser Bereich ist meist nicht größer als ein „Double Word", wodurch er in einem Testzyklus beschrieben werden kann.

Eine Prüfung auf Übertragungsfehler im Rahmen dieses Tests kann nicht durchgeführt werden. Dies liegt daran, dass die Lage des Speicherbereichs im Vergleich zu den anderen Bereichen von SPS zu SPS variiert. Ist dessen Lage im konkreten Einzelfall bekannt, kann auch dann nur eine Prüfung auf

Übertragungsfehler vorgenommen werden, wenn der Bereich der globalen Variablen an diesen Bereich angrenzt.

Wenn der Speichertest mittels Zeiger aus Kapitel 10.6.3 eingesetzt wird, der Übertragungsfehler im gesamten RAM entdecken kann, so sollte versucht werden auch den Test dieses Kapitels durch den Speichertest mittels Zeiger abzudecken.

10.7.1.2 Diagnosedeckungsgrad

Die Effizienz des Testes liegt wie beim Test aus Kapitel 10.6.1 bei 60%.

10.7.2 T8a) Test der redundanten Eingänge

Im Folgenden wird lediglich das Konzept eines solchen Tests beschrieben. Die Programmierung selbst ist nicht Teil dieser Diplomarbeit. Siehe hierzu auch „Ausblick“ in Kapitel 13.3.4.1.

10.7.2.1 Erläuterung

Die Eingangssignale werden in diesem Test parallel über je zwei gleiche Eingänge eingelesen (siehe Abbildung 62). Ein Funktionsbaustein überwacht die von beiden Eingängen an den Prozessor weiterleiteten Signale und prüft diese auf Gleichheit. Zur Optimierung dieses Prozesses sollten die o. a. Eingänge auf verschiedenen Eingangskarten (siehe Abbildung 30) liegen, da parallele Eingänge einer Eingangskarte in Teilen der Signalverarbeitung dieselbe Hardware der Karte benutzen (siehe Abbildung 29).

Zu berücksichtigen ist bei diesem Test, dass es auf Grund der Trägheit des Gesamtsystems - Kabelwege, Eingangskarten - bei der Signalverarbeitung der zu vergleichenden Einzelsignale zu Verzögerungen kommen kann. Aus diesem Grund wird ein Fehler nur dann angenommen, wenn die Eingänge für längere Zeit unterschiedliche Signale ausgeben.

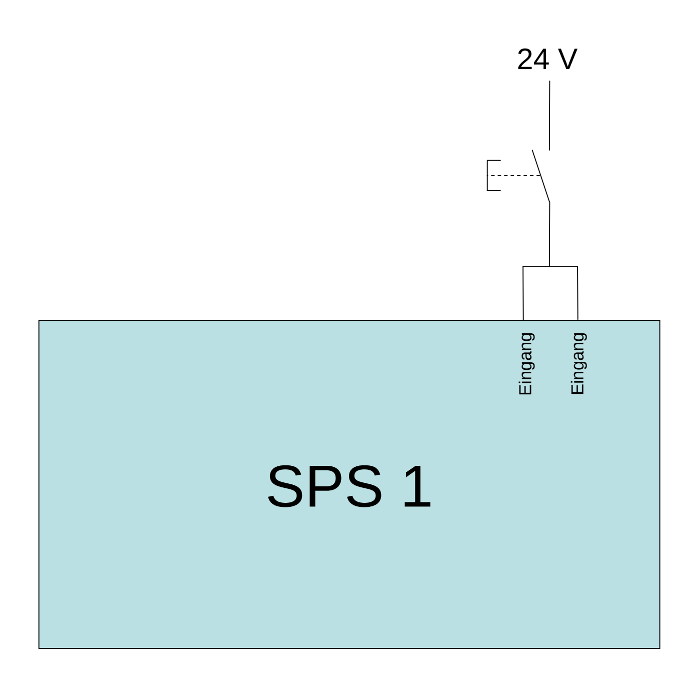


Abbildung 62: Redundante Eingänge

10.7.2.2 Algorithmus

Programmtechnisch kann dies gelöst werden, indem alle verknüpften Eingänge verglichen werden und bei Ungleichheit der Eingänge eine Zählvariable inkrementiert wird. Erreicht diese Zählvariable nach mehreren Programmzyklen einen vorgegebenen Wert, wird der Fehlerbaustein ERROR_FB (siehe Kapitel 10.2) aufgerufen und damit die SPS angehalten wodurch der gesamte von der SPS gesteuerte Prozess in den sicheren Zustand geht.

10.7.2.3 Diagnosedeckungsgrad

Die Effizienz dieses Tests ist nur gering, da Fehler, die einen Eingang blockieren solange nicht entdeckt werden, bis dieser Eingang angesprochen wird. Der DC liegt bei ca. 60 %, in den Fällen, in denen die Eingänge selten angesprochen werden, und bei 90% wenn die Eingänge im Betrieb häufig angesprochen werden. (siehe Tabelle A.7 der DIN EN 61508-2 [17]) Fällt ein Eingang in der Zeit aus, in der er nicht benutzt wird, so kann der Ausfall des zweiten Eingangs in dieser Zeitspanne zu einem so genannten Doppelfehler führen, der durch diesen Test nicht erkannt wird. Aus diesem Grund ist zu Berücksichtigen, dass die Wahrscheinlichkeit eines Doppelfehlers mit zunehmender Zeitspanne zwischen den Betätigungen steigt.

Um solche Doppelfehler auszuschließen sollte wie oben beschrieben möglichst unabhängige Hardware an den Eingängen verwendet werden, um die Fehler durch gemeinsame Ursachen, z. B. den Ausfall eines Optokopplerbausteines, zu minimieren. Dazu können die Eingänge wie beschrieben auf verschiedene Eingangskarten verteilt werden.

10.7.3 T8b) Test der redundanten Ausgänge über rückgekoppelte Eingänge

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	OO_TEST
Namen der Nebenfunktionen	OO_TEST_FALSE OO_TEST_TRUE
Kurzbeschreibung	Test der redundanten Ausgänge über rückgekoppelte Eingänge.
Eingänge	
keine	
Ausgänge	
keine	
Genutzte globale Variablen	
Speicherstellen der Ein- und Ausgänge	

Output_2_3	Diese Variable wird vom Programmierer anstatt der eigentlichen Ausgangsvariable angesprochen. Die eigentlichen Ausgangsvariablen (hier 2 und 3) werden vom Programm verwaltet.

10.7.3.1 Erläuterung

Dieser Test setzt voraus, dass die Ausgänge wie in Abbildung 63 mit Eingängen verknüpft wurden. Diese Verknüpfung erlaubt es, wie in Kapitel 9.2.7 ausführlich beschrieben, das von den Ausgängen ausgegebene Signal über die Eingänge zu überwachen.

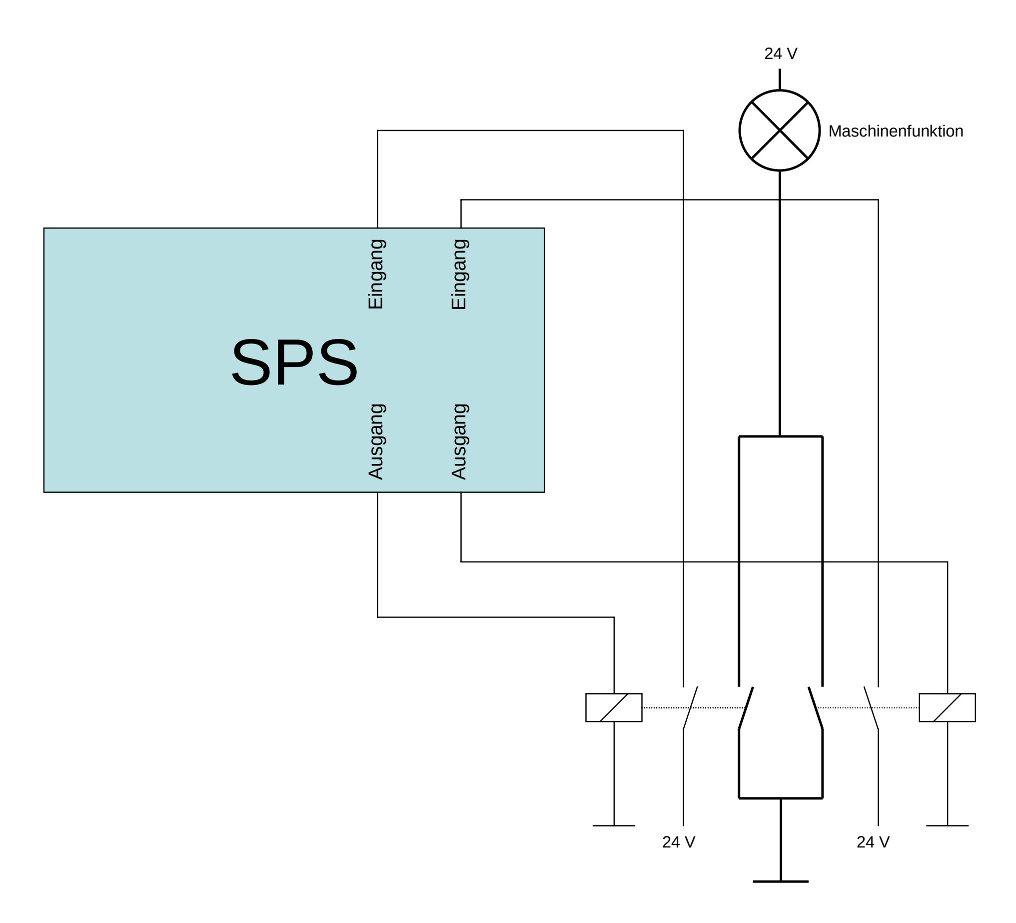


Abbildung 63: Oder-Schaltung redundanter Ausgänge, die rückgelesen werden

Das Testprogramm prüft alle miteinander verschalteten Ausgänge in aufeinander folgenden Zyklen.

Der Test eines redundanten Ausgangspaares ist abhängig davon, ob der durch das Paar erzeugte Ausgang ein- (Maschine läuft) oder ausgeschaltet (Maschine steht) sein soll.

Im laufenden Betrieb der Maschine dürfen beim Testen der Ausgänge nicht beide Ausgänge gleichzeitig ausgeschaltet werden, da die Maschine sonst stoppen würde. Deshalb werden im laufenden Betrieb die Ausgänge nacheinander abgeschaltet und wieder eingeschaltet, so dass immer nur ein Ausgang zur Zeit abgeschaltet ist. Die Schaltzustände der Ausgänge werden über die mit ihnen verknüpften Eingänge überprüft.

Steht die Maschine, d. h. der Ausgang ist abgeschaltet, wird auch dies anhand der verknüpften Eingänge kontrolliert. Allerdings kann in diesem Zustand nicht getestet werden, ob das Einschalten

der Ausgänge funktioniert, da die Maschine dann ungewollt anlaufen würde. Würden die Ausgänge in einer UND-Schaltung verknüpft, so könnte der Test der Ausgänge auf Ein- und Abschalten im ausgeschalteten Zustand aber dann nicht mehr im eingeschalteten Zustand getestet werden. Da der kritische Zustand normalerweise beim fehlenden Abschaltvermögen vorliegt, ist es wichtiger die Abschaltfunktion zu überwachen, weswegen in dieser Diplomarbeit die beschriebene Schaltungsanordnung gewählt wurde.

In beiden Fällen kann der Anwender angeben, wie viele Zyklen zulässig sind, bis das an den Eingängen empfangene Signal dem Signal der verknüpften Ausgänge gleichen muss.

Die zu testenden Ausgänge dürfen während des Anwenderprogramms nicht direkt verändert werden, da dies die Testergebnisse verfälschen würde. Deshalb muss der Programmierer die vom Anwenderprogramm geforderten Ausgangswerte in einer globalen Variablen zwischenspeichern, die erst über das Testprogramm auf den wirklichen Ausgangsspeicher weitergeleitet wird. Aus diesem Grund ist das Testprogramm am Ende des Programmzyklus aufzurufen, da es so den gewünschten Wert vor dem eigentlichen Setzen der Ausgänge schreiben kann.

10.7.3.2 Algorithmus

Der Gesamttest gliedert sich auf in

- eine Hauptfunktion, die die eigentlichen Tests in den zwei Nebenfunktionen vorbereitet und aufruft,
- zwei Nebenfunktionen

Die Hauptfunktion ist in Abbildung 64 im Flussdiagramm dargestellt und im nachfolgenden Text beschrieben.

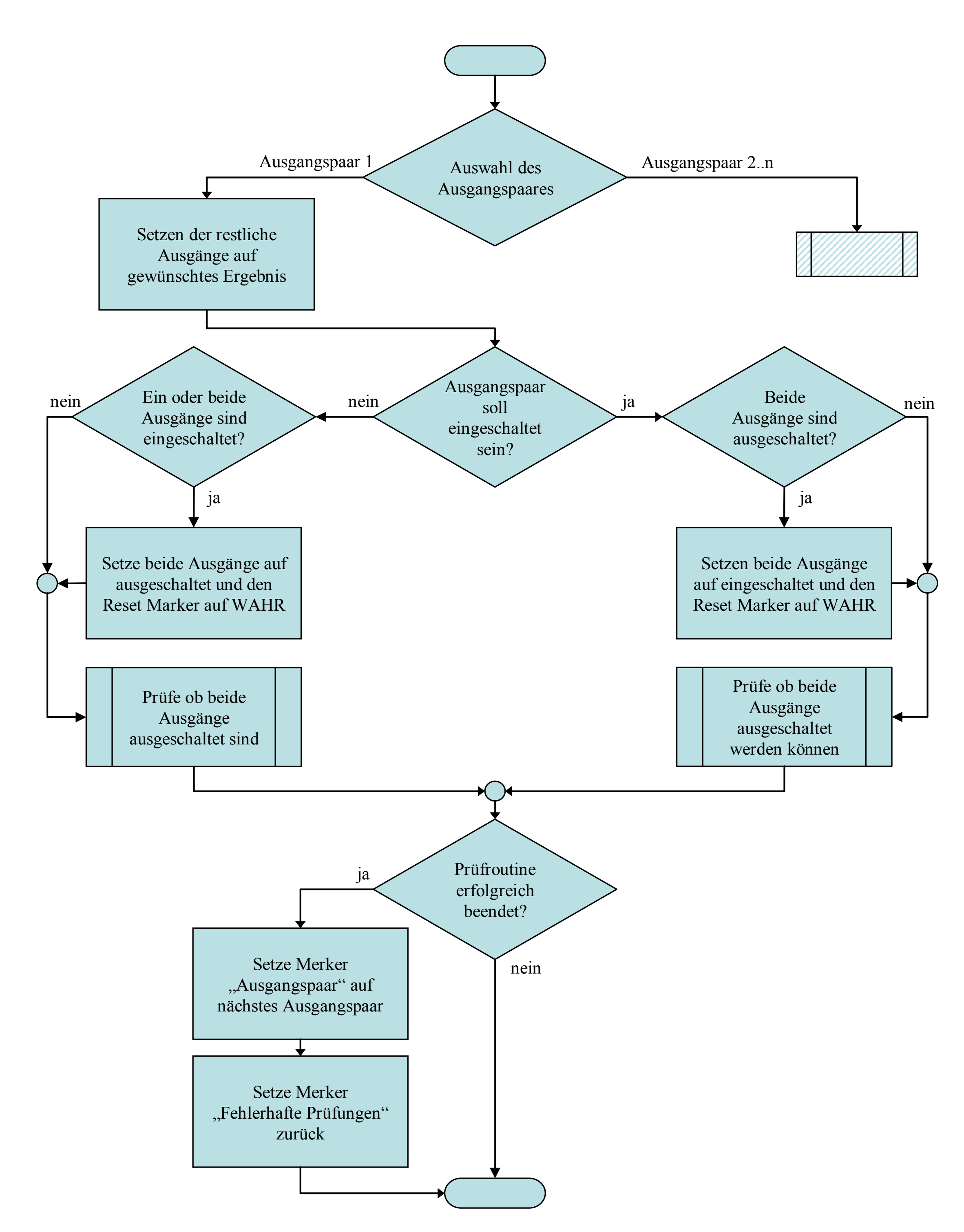


Abbildung 64: Flussdiagramm Test der Ausgänge

Nach der Auswahl des zu testenden Ausgangspaares werden die Daten des Anwenderprogramms an die restlichen nicht zu testenden Ausgänge weitergeleitet.

Danach folgt eine Abfrage, ob die Ausgänge des zu testenden Ausgangspaares ein- oder ausgeschaltet sein sollen. Die beiden daraus resultierenden Programmabläufe werden nachfolgend beschrieben.

Erster Fall, das Ausgangspaar soll ausgeschaltet sein:

Im Speicherbereich beider Ausgänge wird geprüft, ob die Ausgänge bereits auf den Zustand „ausgeschaltet“ gesetzt sind. Ist dies nicht der Fall, bedeutet das, dass der Test in diesem Zyklus

beginnt, da vorher das Ausgangspaar eingeschaltet war. Daher müssen beide Ausgänge auf „ausgeschaltet“ gesetzt werden und die Zähler im entsprechenden Testbaustein OO_TEST_FALSE (s. u.) zurückgesetzt werden. Waren beide Ausgänge bereits vorher auf „ausgeschaltet“ gesetzt, so wird dieser Punkt übergangen, da die Ausgänge bereits in einem vorhergehenden Zyklus abgeschaltet wurden.

Nach dieser Abprüfung wird in beiden Fällen ein Test aufgerufen, der prüft ob die verknüpften Eingänge bestätigen, dass die Ausgänge auch nach außen hin ausgeschaltet sind. (siehe Kapitel 9.2.7)

Zweiter Fall, das Ausgangspaar soll eingeschaltet sein:

Im Speicherbereich beider Ausgänge wird geprüft, ob die Ausgänge auf den Zustand „ausgeschaltet“ gesetzt sind. Ist dies der Fall, bedeutet das, dass Ausgangspaar im aktuellen Zyklus erst eingeschaltet wird. Daher müssen auch hier die Zähler im entsprechenden Testbaustein OO_TEST_TRUE (s. u.) zurückgesetzt werden. Ist nur einer der beiden Ausgänge eingeschaltet, bedeutet das, dass die Prüfroutine bereits im letzten Zyklus gelaufen ist. In diesem Fall wird hier keine Veränderung vorgenommen, da dies die Prüfroutine, die immer über mehrere Zyklen läuft, stören würde.

Nach Abprüfung der aktuellen Zustände an den Ausgängen wird in beiden Fällen der Testbaustein OO_TEST_TRUE aufgerufen, der prüft ob die verknüpften Eingänge bestätigen, dass die Ausgänge auch nach außen hin eingeschaltet sind und ausgeschaltet werden können.

Welcher der beiden Testbausteine aufgerufen wurde, ist im weiteren Verlauf unwichtig, da nur noch geprüft werden muss, ob die aufgerufene Routine erfolgreich war. Diese Prüfung ist in beiden Fällen dieselbe.

Eine komplette Prüfung der Ausgänge benötigt mehrere Zyklen. Deshalb kann aus einer nicht erfolgreichen Prüfung innerhalb eines Zyklus noch nicht auf einen Fehler geschlossen werden. Eine Reaktion auf zu häufige Fehlprüfung geschieht in den jeweiligen Testroutinen (s. u.), weswegen das Hauptprogramm hierauf nicht reagieren muss.

Im Falle einer erfolgreich beendeten Prüfung wird der Zähler für die Ausgangspaare auf das nächste Ausgangspaar gesetzt, und ein Merker gesetzt, der der Testroutine des nächsten Zyklus mitteilt, dass Sie einen neuen Test beginnt und somit die internen Zähler zurücksetzen muss.

Die Nebenfunktionen bestehen aus dem Testbaustein OO_TEST_TRUE und dem Testbaustein OO_TEST_FALSE. Von diesen Testbausteinen wird im Folgenden nur der Baustein OO_TEST_TRUE beschrieben, dessen Flussdiagramm in Abbildung 65 dargestellt ist. Der Testbaustein OO_TEST_FALSE prüft lediglich die Gleichheit von den Ein- und Ausgängen und zählt in Zyklen, in denen diese ungleich sind, eine Variable hoch. Erreicht diese Variable einen vom Programmierer festgelegten Wert, bevor die Gleichheit der Ein- und Ausgänge festgestellt werden kann, so wird der Fehlerbaustein ERROR_FB aufgerufen.

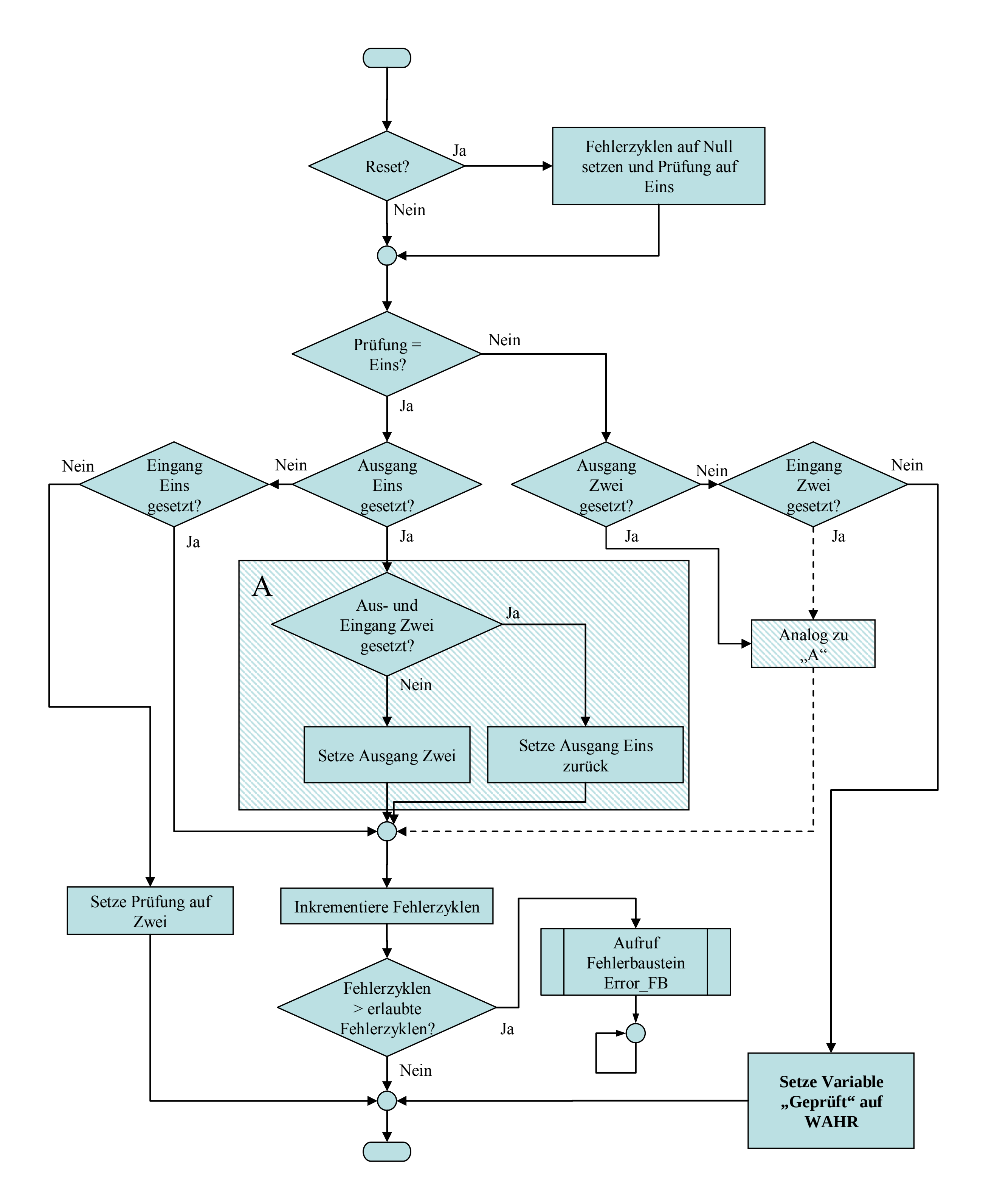


Abbildung 65: Flussdiagramm des Testbausteins OO_TEST_TRUE – Prüfe ob beide Ausgänge ausgeschaltet werden können

Als erstes wird in der Testroutine OO_TEST_TRUE bei ihrem Aufruf überprüft, ob die Laufvariable, die die Fehlerzyklen zählt, sowie die die sich den zu prüfenden Ausgang des Paares merkt, zurückgesetzt werden soll („Reset“). Das bedeutet in diesem Fall wird der Zähler für durchlaufene Prüfungen auf Null gesetzt und der Merker für die Prüfung wird auf den ersten Ausgang des Ausgangspaares gesetzt.

Nach der Auswahl des zu Prüfenden Ausgangs (Eins oder Zwei) laufen beide Tests, die des ersten und die des zweiten Ausgangs, weitestgehend analog ab:

In beiden Fällen wird geprüft, ob der zu prüfende Ausgang gesetzt ist. Sollte dies der Fall sein, wird er abgeschaltet, sobald sichergestellt ist, dass der andere Ausgang sicher gesetzt ist. Dies überprüft einerseits das Einschaltverhalten des jeweils anderen Ausgangs, andererseits wird so sichergestellt, dass es durch die Prüfung zu keinem ungewollten Abschalten der durch die SPS gesteuerten Maschine kommt.

Ist der zu prüfende Ausgang nicht gesetzt wird überprüft, ob sein verknüpfter Eingang dies bestätigt. Eine erfolgreiche Prüfung des ersten Ausgangs führt zum Setzen des Merkers der Prüfungen auf den Zweiten Ausgang. Eine erfolgreiche Prüfung des zweiten Ausgangs führt zum erfolgreichen Beenden des Tests und damit zum Setzen der Variablen *Geprüft* auf WAHR, die wiederum von der Hauptfunktion gelesen und verarbeitet wird (s. o.).

Jeder Aufruf der Routine, der nicht zu einem Erfolg oder Teilerfolg führt wird als Fehlerzyklus gezählt. Überschreitet die Anzahl der Fehlerzyklen die vom Programmierer festgelegte Anzahl an erlaubten Fehlerzyklen, so wird der Fehlerbaustein ERROR_FB aufgerufen.

10.7.3.3 Diagnosedeckungsgrad

Die Effizienz dieses Tests liegt bei größer gleich 99%. Das Ein- und Ausschaltverhalten der Ausgänge kann zwar nur im eingeschalteten Zustand geprüft werden, das Ausschalten unterliegen aber einer ständigen Überwachung.

Kommt es zu einem Fehler an einem der beiden Ausgänge oder nach geschalteten Relais, der das Abschalten verhindert, so kann es zu einem kritischen Zustand kommen. Der Stromverlauf kann dann durch die SPS nicht mehr unterbrochen werden, obwohl der Fehler bemerkt wird. Dieses Problem wird in Kapitel 0 dieser Diplomarbeit aufgegriffen und eine Lösung mittels eines redundanten Systems durch zwei SPSen vorgestellt.

Zusätzlich zur Fehlererkennung des mit diesem Test getesteten Ausgangs wird hierdurch auch ein Fehler im verknüpften Eingang festgestellt. Die durch diesen Test getesteten Eingänge sollten gleichmäßig auf alle Eingangskarten verteilt werden um somit alle Eingangskarten auf Fehler in jeweils gemeinsam genutzten Bauteilen zu testen. Siehe hierzu auch „Ausblick“ in Kapitel 13.3.4.2. Der DC des in diesem Kapitel beschriebenen Tests ist auf die Eingänge bezogen allerdings geringer als der DC des im „Ausblick“ beschriebenen Tests, da hier kein regelmäßiges dynamisches Signal eingesetzt wird. Der konkrete DC ist von der Anzahl der gemeinsam genutzten Bauteile abhängig und kann aus diesem Grund nicht allgemeingültig angegeben werden.

10.8 Fehlererkennung durch gegenseitige Überwachung mit einer zweiten SPS

Alle vorgenannten Tests beziehen sich auf die sicherheitstechnische Verbesserung einer einzelnen Standard-SPS. Diese sicherheitstechnische Verbesserung hat durch die „Einkanaligkeit“ des Systems jedoch Grenzen. Eine weitere Verbesserung kann mit einer externen Testeinrichtung verwirklicht werden, die einen separaten Abschaltpfad für das System besitzt. Das bedeutet, die Testeinrichtung kann den von der SPS geschalteten Strompfad unterbrechen.

In dieser Diplomarbeit wurde als Testeinrichtung eine weitere SPS eingesetzt, die grundsätzlich ebenfalls mit den bisher erwähnten Selbsttests ausgestattet ist. Eine Abweichung hinsichtlich des Tests der Ausgänge wird in Kapitel 0 beschrieben. Die Möglichkeiten zur Verschaltung der beiden SPSen sind in Kapitel 10.8.1 dargestellt. Die sicherheitstechnischen Verbesserungen durch den Einsatz der zweiten SPS sind wie folgt:

- Fällt eine SPS so aus, dass ein Abschalten nicht mehr möglich ist (Relais „klebt“ oder Fehler im Watchdog), so kann dies durch die zweite SPS durchgeführt werden.
- Mit Hilfe einer zweiten SPS ist es möglich bestimmte in der ersten SPS auftretende Fehler zu entdecken (z. B. Fehler im Takt der SPS).
- Wie in Kapitel 12 gezeigt wird, ist diese zweite SPS für das Erreichen der Kategorie 2 nach DIN EN 954-1 zwar nicht nötig, wohl aber für das Erreichen der Kategorie 2 nach prEN 13849-1. Der Einsatz einer Testeinrichtung in einer Kategorie 2 Steuerung wird schon heute vom BGIA in der Quelle [6], einem Leitfaden zur DIN EN 954-1, vorgeschlagen.

10.8.1 Schaltungsmöglichkeiten

In Abbildung 66 wird die Verschaltung von zwei SPSen mit einem redundanten Abschaltpfad dargestellt. Die Abschaltwege sind so verknüpft, dass bei einem Fehler in einem Relais einer der beiden SPSen ein Abschalten der Maschinenfunktion weiterhin möglich ist.

Die obere SPS 1 ist mit dem Programm zur Steuerung der eigentlichen Funktion und den in der Diplomarbeit beschriebenen Selbsttests programmiert. Die untere SPS 2 beinhaltet nur die in der Diplomarbeit beschriebenen Selbsttests. Die untere SPS 2 kann sich dabei wesentlich intensiver selbst testen, da sie keine Rechenzeit für ein ablaufendes Programm benötigt.

Die ablaufenden Selbsttests der beiden SPSen kommunizieren wie in Kapitel 0 beschrieben über einen Feldbus miteinander, um ein Ausfallen der jeweiligen anderen SPS zu bemerken.

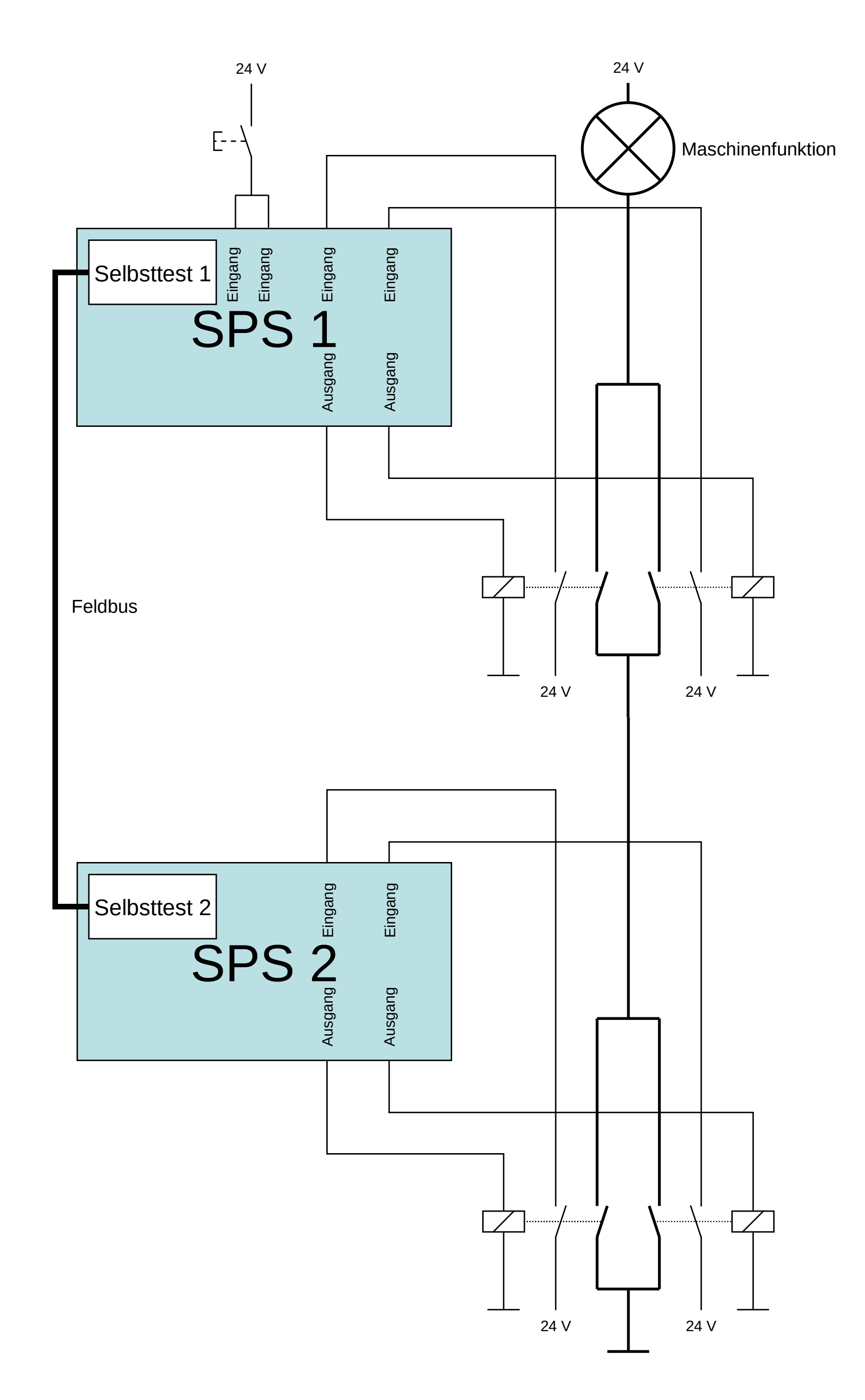


Abbildung 66: Verschaltung von zwei SPS zu einem redundanten Abschaltweg

In Abbildung 67 wird eine Variante der Verschaltung von zwei SPSen mit einem redundanten Abschaltpfad dargestellt. Die Verschaltung in Abbildung 67 spart dabei ein gekoppeltes Relais ein. Je nach zu schaltender Stromstärke ist dies ein großer Kostenfaktor.

Durch das fehlende Relais ist die SPS 1 allerdings nicht mehr in der Lage einen präventiven Selbsttest durchzuführen. Dies verringert die Testgüte bei geringen Schaltspielen, da die Wahrscheinlichkeit eines Doppelfehlers im System und der Testeinrichtung („kleben bleiben" jeweils eines Relais) steigt, auf den dann nicht mehr reagiert werden kann. Der in SPS 1 eingesetzte Test vergleicht nur noch den Soll- mit dem Ist-Zustand.

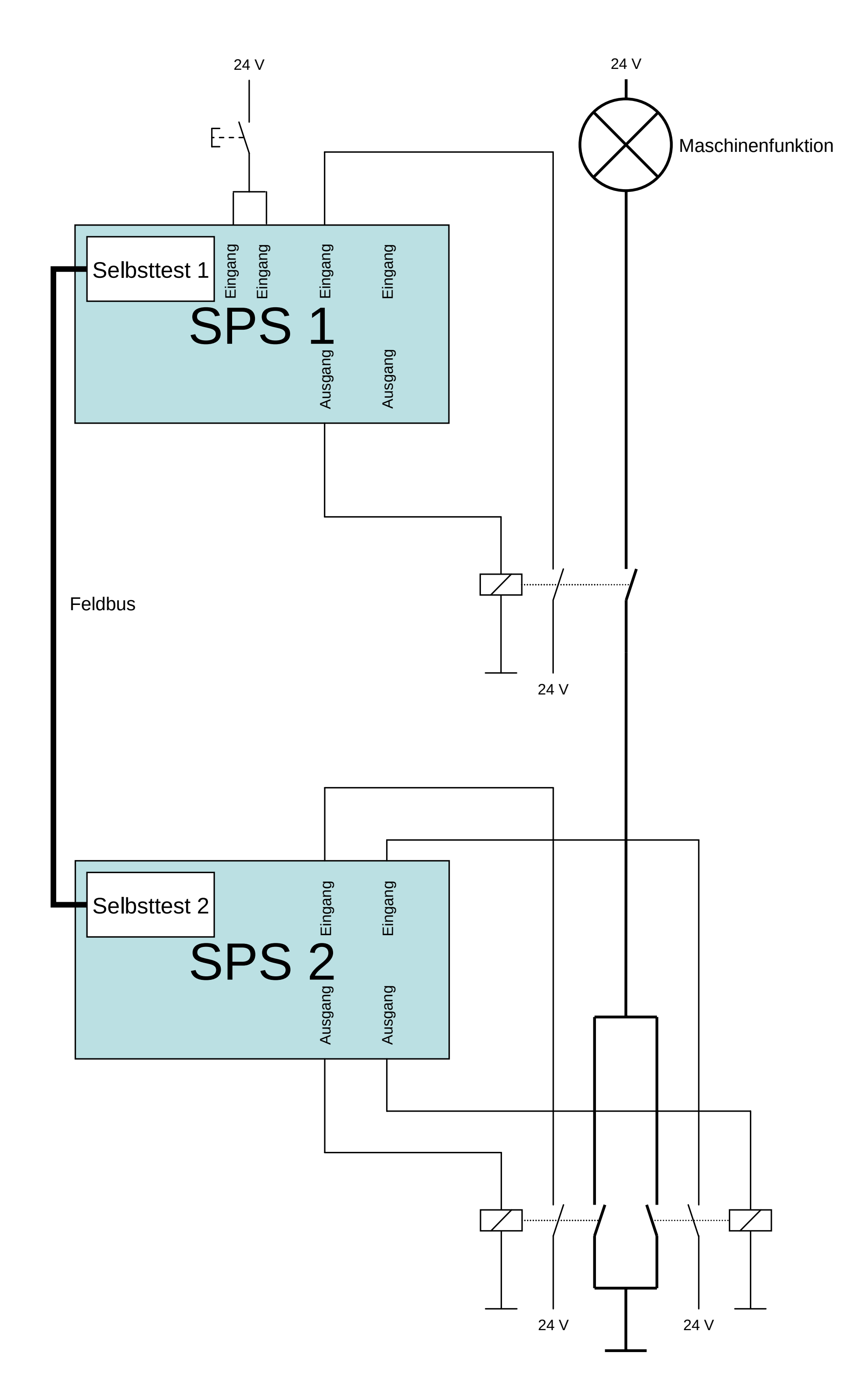


Abbildung 67: Verschalten von zwei SPS zu einem Abschaltweg

10.8.2 T2b) Gegenseitige Funktionskontrolle

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	ABLAUF_UEBERWACHUNG_SEND
Namen der Nebenfunktionen	ABLAUF_UEBERWACHUNG_INIT
Kurzbeschreibung	Konstantes Austauchen von Zahlen um der anderen SPS mitzuteilen, dass die eigene SPS noch lebt.
Eingänge	
Equals_till_Error	Anzahl, wie oft die gleiche Zahl von der anderen SPS in Folge empfangen werden darf.
ListenStatus_last_differenz_Max	Maximaler Zahlenunterschied zwischen der aktuellen empfangenen Zahl und der zuletzt geprüften Zahl.
Ausgänge	
andereSPSläuft	Anzeige, ob beide SPSen bereit sind, den Test zu starten. Dies ist nötig, damit eine SPS die zuerst eingeschaltet wird keinen Fehler meldet weil sie nichts empfängt.
Genutzte globale Variablen	
SendStatus	Speicherstelle zum Absenden von Informationen
LastSend	Zuletzt gesendete Nummer
ListenStatus	Speicherstelle zum empfangen von Informationen
ListenStatus_last	Zuletzt empfangene Information

10.8.2.1 Erläuterung

Bei der Verwendung einer einzelnen Standard-SPS mit Diagnosefunktionen können Fehler auftreten, die nicht bemerkt werden bzw. zwar bemerkt werden aber auf die nicht reagiert werden kann. Diese Fehler können dazu führen, dass die SPS nicht abschalten kann. Dadurch kann ein gefährlicher Zustand herbeigeführt werden. Die verhinderte Abschaltfunktion kann von einer zweiten SPS übernommen werden. (siehe Kapitel 10.8) Um sicherzustellen, dass die zweite SPS nicht selbst bereits durch einen Fehler ausgefallen ist, muss auch diese überwacht werden. Dies kann von der ersten SPS übernommen werden. In diesem Test dargestellt, wie eine gegenseitige Überwachung zweier SPSen realisiert werden kann.

Dies wird über zwei Variablen erreicht, über die sich die SPSen über einen Feldbus austauschen. Da dieser Feldbus Fehler aufweisen kann, dürfen sich die überprüften Variablen nur in einem vorgegebenen Rahmen ändern. Dieser Datenaustausch über den Feldbus benötigt in der Regel einige Zyklen, da z. B. die SPS die nur zum Testen eingesetzt wird, eine wesentlich kürzere Zyklusdauer besitzen kann. Dies ist u. a. abhängig von der Aufteilung der hierin ablaufenden Testfunktionen. Aus diesem Grund muss hier eine Toleranz zugelassen werden, die allerdings möglichst gering sein sollte.

10.8.2.2 Algorithmus

Der Test läuft auf beiden SPSen parallel ab. Er besteht in beiden SPSen aus den gleichen Funktionsbausteinen. Einem Funktionsbaustein zur Initialisierung der Kommunikation (ABLAUF_UEBERWACHUNG_INIT) und einem zweiten Funktionsbaustein (ABLAUF_UEBERWACHUNG_SEND), der nach erfolgter Initialisierung die eigentliche Kommunikation durchführt. Der Algorithmus dieser Kommunikation wird in Abbildung 68 dargestellt und im Text erläutert.

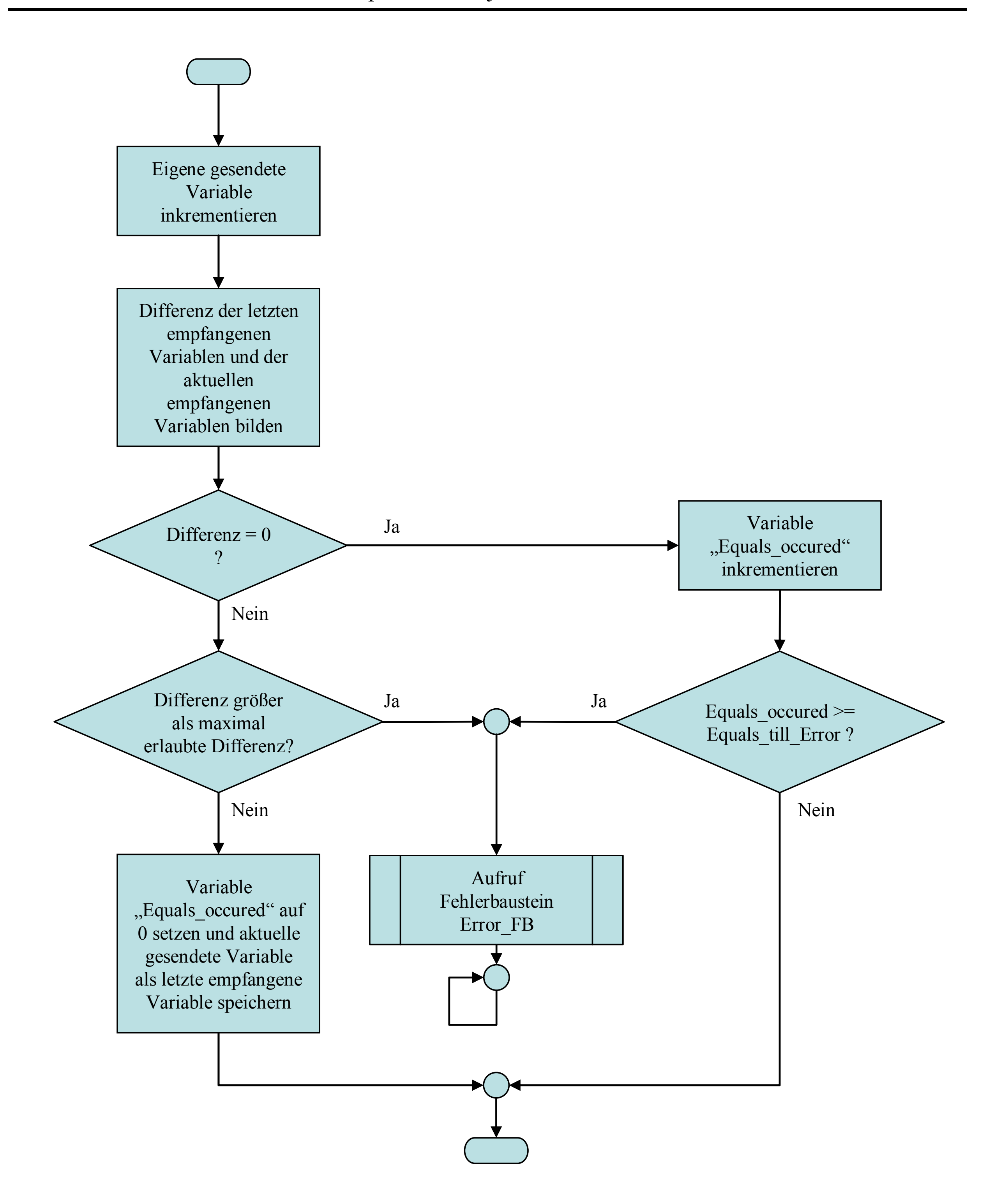


Abbildung 68: Flussdiagramm Sende- und Empfangseinheit Ablaufüberwachung mit zwei SPS

Der Funktionsbaustein ABLAUF_UEBERWACHUNG_SEND inkrementiert bei seinem Aufruf als erstes die Variable *LastSend* und schickt diese über den Feldbus an die jeweils andere SPS.

Danach wird für die weitere Prüfung die Differenz zwischen der zuletzt und der aktuell empfangenen Variablen der jeweils anderen SPS gebildet. Durch die Differenzbildung wird ein Fehler bei einem Überlauf der Variablen am Ende ihres Wertigkeitsbereich (32 bit – 0..4.294.967.296) verhindert.

Hiernach wird zwischen zwei Fällen unterschieden:

Fall Eins, die errechnete Differenz ist Null:

Ist die Differenz Null, also die gesendete Variable seit dem letzten Auslesen nicht geändert worden, so wird eine Testinterne Variable *Equals_occured* inkrementiert, um diese Zyklen zu zählen. Diese Variable wird danach überprüft, ob sie einen vorgegebenen Wert überschritten hat. Dieser vorgegebene Wert ist abhängig von dem Zeitunterschied zwischen den Zyklen der beiden SPSen und der Geschwindigkeit des Datenaustausches. Im konkreten Einzelfall muss dieser Wert empirisch ermittelt werden, so dass er sicher keinen Fehlalarm auslöst. Es ist hierbei zu beachten, dass zwar der DC des Tests durch den Wert nicht beeinflusst wird, die Trägheit des Systems aber mit zunehmender Größe dieses Wertes steigt. (siehe hierzu Echtzeitfähigkeit in Kapitel 6.5)

Wurde dieser vom Anwender festgelegte Wert überschritten, so wird der Fehlerbaustein ERROR_FB aufgerufen und damit die SPS abgeschaltet. Andernfalls wird der Testzyklus verlassen.

Fall Zwei, die errechnete Differenz ist ungleich Null:

Ist die Differenz ungleich Null, so wird überprüft, ob Sie unter einer maximal zulässigen, vom Anwender festgelegten, Größe liegt. Diese Größe stellt die Anzahl der Zyklen dar, die die jeweils andere SPS durchlaufen kann, bis dieser Testbaustein erneut aufgerufen wird. Sie muss, wie auch die Variable des ersten Falls empirisch ermittelt werden. Sie beeinflusst ebenfalls nicht den DC, sondern auch hier nur die Echtzeitfähigkeit des Systems.

Wenn die Differenz über der o. a. festgelegten Größe liegt, wird der Fehlerbaustein ERROR_FB aufgerufen und damit die SPS abgeschaltet. Liegt sie im zulässigen Bereich ist die Prüfung erfolgreich abgeschlossen. In diesem Fall wird die Variable *Equals_occured* (Zähler für Null-Differenzen für Fall Eins) zurückgesetzt, sowie die aktuell empfangene Variable als zuletzt empfangene Variable abgespeichert.

10.8.2.3 Diagnosedeckungsgrad

Die Effizienz für die Überwachung der Funktonalität der jeweils anderen SPS liegt bei größer gleich 99 %, da sich beide SPSen in jedem Zyklus gegenseitig überwachen.

Durch diesen Test werden ebenfalls Fehler im Feldbus aufgedeckt. Auch hier liegt der DC bei 99%, da das Fenster für die zu empfangenden Variablen gemessen an der Größe des möglichen Zahlenbereichs sehr gering ist. Selbst bei einer vom Programmierer zugelassenen Abweichung von 30 Zyklen in Fall Zwei des vorhergehenden Kapitels stehen diese im Verhältnis von 30 zu 4.294.967.296 möglichen Werten.

10.8.3 T8c) Test der redundanten Ausgänge über rückgekoppelte Eingänge für die zweite SPS

Die nachfolgende Tabelle enthält die Bezeichnungen der für diesen Test verwendeten Funktionen und Variablen:

Name der Hauptfunktion	OO_TEST_ZWEITE_SPS
Namen der Nebenfunktionen	OO_TEST_TRUE
Kurzbeschreibung	Dauertest der redundanten Ausgänge über rückgekoppelte Eingänge.
Eingänge	
keine	
Ausgänge	
keine	
Genutzte globale Variablen	
keine	

10.8.3.1 Erläuterung

Dieser Test basiert auf dem in Kapitel 10.7.3 beschriebenen Test und wird in der zweiten SPS an dessen Stelle eingesetzt. Da die zweite SPS dauerhaft eingeschaltet ist, entfällt hier eine Überprüfung der „gewünschten Zustände" an den Ausgängen.

Da der Algorithmus des Funktionsbausteins OO_TEST_TRUE bereits im zweiten Teil des Kapitels 10.7.3.2 beschrieben wurde, und die Hauptfunktion desselben Kapitel nur wie in Abbildung 69 zu erkennen gekürzt wurde, wird an dieser Stelle auf eine Beschreibung des Algorithmus verzichtet.

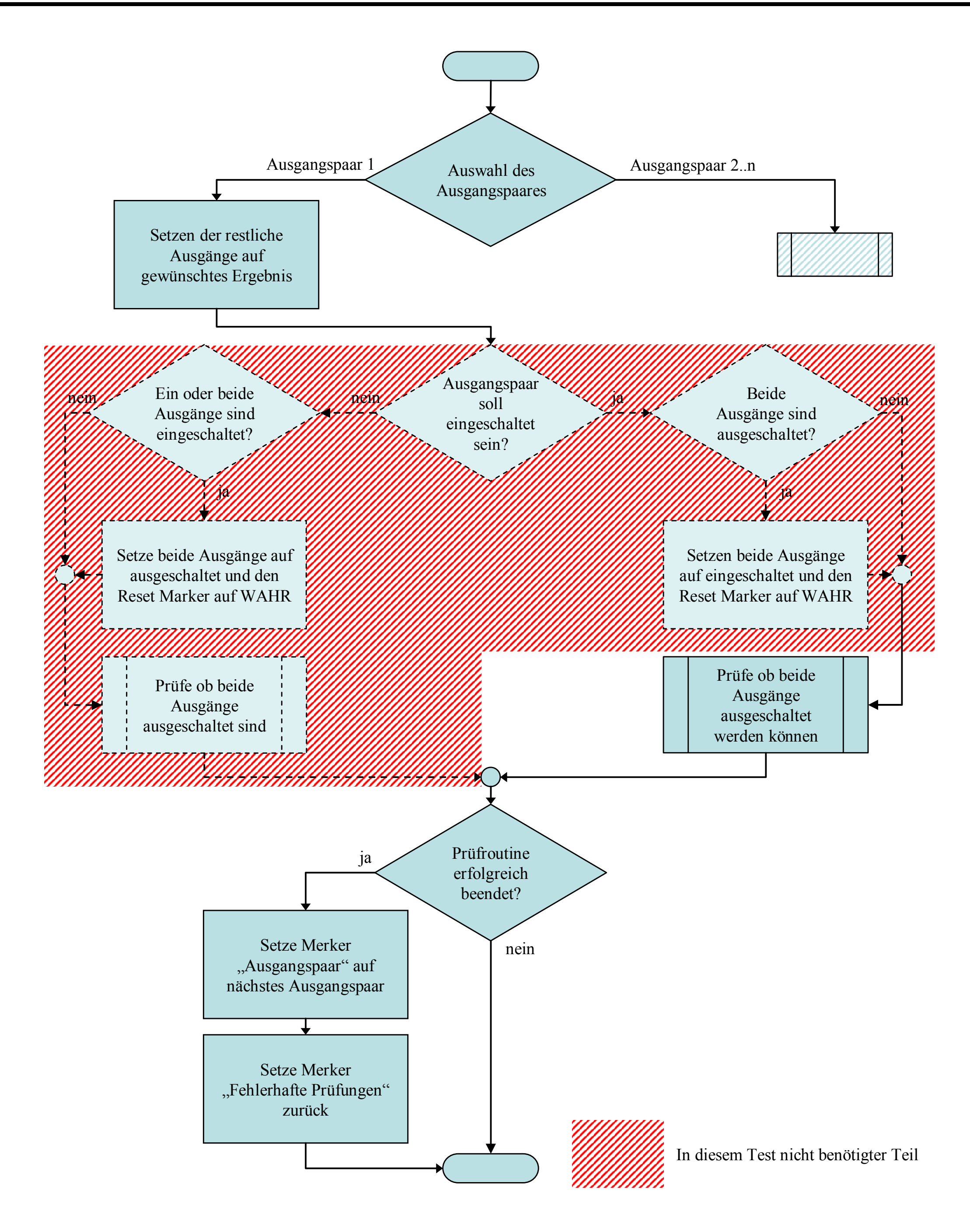


Abbildung 69: Flussdiagramm der Hauptfunktion für das Prüfen der Ausgänge der zweiten SPS (vgl. Abbildung 64 für erste SPS)

10.8.3.2 Diagnosedeckungsgrad

Die Effizienz steigert sich zwar durch die häufigere Überprüfung geringfügig gegenüber dem Test aus Kapitel 10.7.3. Da die ursprüngliche Testroutine bereits einen DC von 99% erreicht, hat dieser Test auch einen DC von 99%.

11 Übersicht über mögliche Hardware-Probleme und deren Lösungen

Die nachfolgende Tabelle enthält eine Übersicht über die möglichen Fehler in einer SPS. Zugeordnet zu diesen Fehlern sind jeweils der in dieser Diplomarbeit betrachtete allgemeine Ansatz zum Erkennen ähnlicher Probleme in anderen Bereichen, sowie der Testbaustein zum Auffinden der Fehler in einer SPS (in dieser Diplomarbeit entwickelt / herstellerseitig implementiert) und sein DC (wie in prEN ISO 13849-1 definiert).

Fehler aus Kapitel 9.1	**Kapitel:** **Übliche Lösung**	**Kapitel:** **Lösung für die SPS**	**Erreichter DC**
F1 – Takt	L1) 9.2.1	T1a) 10.3.1	50%
F2 – Programmzähler	L2) 9.2.2	T2a) 10.3.2 T2b) 10.4.2.1	90 – 99% 60%
F3 – Akkumulator	L3) 9.2.3	T3) 10.4.2.2	90%
F4 – ALU	L4) 9.2.4	T4a) 10.4.2.3 T4b) 10.4.2.4 T4c) 10.4.2.5	60% 60% 60%
F5 – Kommunikation zwischen Prozessor und Speicher	L5) 9.2.5	T5) 10.4.2.6	60%
F6 – „Stuck at“	L6) 9.2.6	T6-RAM) 10.6.1 T6-EA) 10.7.1	60% 60%
F7 – Übertragsfehler	L7) 9.2.6	L7-ROM) 10.5.1 L7a-RAM) 10.6.2 L7b-RAM) 10.6.3	60 – 90% 90% 90%
F8 – Ein- und Ausgabe	L8) 9.2.7	T8a) 10.7.2 T8b) 10.7.3 T8c) 0	60 – 90% 99% 99%

12 Erreichte Erhöhung der Sicherheit einer Standard SPS

12.1 Einstufung der Standard-SPS mit Diagnosefunktionen nach EN 954-1

Die höchste Kategorie die eine Steuerung nach DIN EN 954-1 [14] erreichen kann, ohne redundant zu sein, ist Kategorie 2. Dies ist aus der Tabelle in Kapitel 5.2.1 in Abbildung 7 ersichtlich. Die Anforderungen aus der genannten Tabelle für Kategorie 2 können erfüllt werden, indem die Tests von Kapitel 10.3 bis einschließlich Kapitel 10.7 umgesetzt werden:

- Fehlererkennung durch Ablaufkontrolle
    - T2a) Selbst programmierte Watchdog Funktion mit Ablaufkontrolle
- Fehlererkennung durch Prozessortest
    - T2b) Test der bedingten Sprünge
    - T3) Test des Akkumulators
    - T4a) Test der Logischen Operatoren
    - T4b) Test der Arithmetischen Operatoren
    - T4c) Test der Komparatoren
    - T5) Test des Ladens und Speicherns von Daten
- Fehlererkennung durch Speichertest des RAM
    - T6-RAM) Test der benutzten Variablen
    - T7a-RAM) Test des Bereichs der globalen Variablen mittels eines Arrays
- Fehlererkennung durch Test der Ein- und Ausgabe
    - T6-EA) Test des Speichers der Ein- und Ausgänge
    - T8a) Test der redundanten Eingänge
    - T8b) Test der redundanten Ausgänge über rückgekoppelte Eingänge

Ist die gewählte SPS kompatibel zu CoDeSys, so können die Speichertests des RAM durch den Test

- T7b-RAM) Test des Speichers mittels eines Zeigers (CoDeSys-spezifisch)

ersetzt werden was zwar zu einer effektiven Verbesserung der Sicherheit, aber zu keiner anderen Kategorie führt.

Das Zuschalten der zweiten SPS, als Testeinrichtung der ersten SPS, ist zum Erreichen der Kategorie 2 nicht nötig. Allerdings führt auch dies zu einer effektiven Verbesserung der Sicherheit und wird deshalb u. a. wie in Abbildung 70 dargestellt vom BGIA vorgeschlagen:

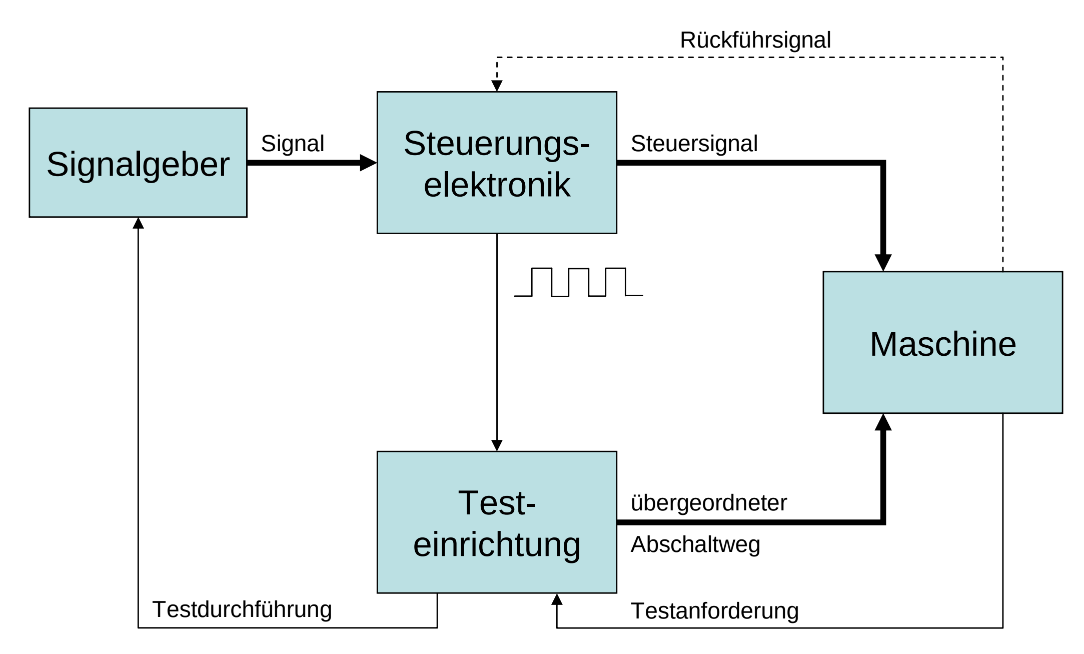


Abbildung 70: Elektroniksteuerung nach EN 954 – Kategorie 2, Abbildung 22 aus BGIA-Report [6]

Die Abbildung 70 zeigt eine Kategorie 2 Steuerung nach DIN EN 954-1 aus dem BGIA-Report [6]. Dies kann umgesetzt werden, indem sämtliche Tests aus Kapitel 10, einschließlich der Verwendung einer zweiten SPS, implementiert werden. Das bedeutet, zu den o. a. Tests kommen noch folgende hinzu:

- Fehlererkennung durch gegenseitige Überwachung mit einer zweiten SPS
    - T2b) Gegenseitige Funktionskontrolle
    - T8c) Test der redundanten Ausgänge über rückgekoppelte Eingänge für die zweite SPS

Auch wenn der Einsatz einer zweiten SPS nicht zu einer Erhöhung der Steuerungskategorie führt, ist dies kein Grund für den Hersteller diese sicherheitstechnische Lösung im konkreten Einzelfall nicht in Betracht zu ziehen. Nach der MRL ist der Hersteller gehalten, eine „ausreichende“ Sicherheit auch hinsichtlich der verwendeten Steuerung zu gewährleisten. Das bedeutet allerdings nicht, dass er immer die teuerste Lösung mit der ggf. technisch möglichen maximalen Sicherheit einsetzen muss.

Der Hersteller muss im Rahmen der rechtlich geforderten Risikobeurteilung ermitteln, welche Steuerung eine „ausreichende“ Sicherheit für seine Maschine bietet. Dabei kann er sich nicht allein auf harmonisierte Normen wie z. B. die DIN EN 954-1 stützen. Ggf. muss er den Stand der Technik ermitteln, der im Einzelfall über die Anforderungen einer harmonisierten Norm hinausgehen kann. In dem Fall der sicherheitstechnischen Steuerungen kann festgestellt werden, dass der Stand der Technik sich für Kategorie 2 Steuerungen über die Anforderungen der fast 10 Jahre alten DIN EN 954-1 hinaus entwickelt hat. Siehe hierzu den BGIA-Report [6] und insbesondere auch die prEN ISO 13849-1 [15], die zukünftig die DIN EN 954-1 ablösen wird.

Beim Umsetzen aller vorgenannten Tests muss sichergestellt werden, dass

- jeder Test zu Beginn komplett durchlaufen wird

  und

- jeder Test während der Betriebszeit periodisch durchlaufen wird.

Beispiele für die Umsetzung dieser Forderung und deren Implementierung in der Programmiersprache AWL sind im Quelltext (Anhang II, Teil CoDeSys) zu finden.

Wird im Rahmen einer Kategorie 2 Steuerung keine zweite SPS verwendet, ist Kapitel 6.2.3 „Kategorie 2“ der DIN EN 954-1 zu beachten:

> *„Jede Prüfung der Sicherheitsfunktion muss […] einen Ausgang für die Einleitung angemessener Steuerungsmaßnahmen erzeugen, wenn ein Fehler erkannt wurde. […] Wenn die Einleitung eines sicheren Zustands nicht möglich ist, z.B. Verschweißen des Kontakts beim Endschalter, muss der Ausgang eine Warnung vor der Gefährdung vorsehen.“ [14]*

Das bedeutet, in diesem Fall ist ein Warnausgang vorzusehen, der im Falle eines fehlerhaften Nicht-Abschaltens durch „klebende“ Relais aktiviert werden muss. Werden wie vorgeschlagen zwei SPSen eingesetzt ist dies dank des zweiten Abschaltweges nicht nötig.

Im gleichen Kapitel der Norm findet sich:

> *Nach Erkennung eines Fehlers muss ein sicherer Zustand bis zur Behebung des Fehlers aufrechterhalten werden. [14]*

Dies wird durch die in Kapitel 10.2 beschriebenen RETAIN Variable *Fehler_erkannt* möglich gemacht. Es muss beim Nutzen der in dieser Diplomarbeit beschriebenen Tests sichergestellt werden, dass diese Variable beim Start der SPS überwacht wird.

12.2 Einstufung der Standard-SPS mit Diagnosefunktionen nach prEN ISO 13849-1

Wie in Kapitel 5.2.3 beschrieben, teilt die prEN ISO 13849-1 die Sicherheitsstufen nach Performance Level (PL) ein und nicht mehr nach Kategorien. Sie benötigt die Einteilung in Kategorien allerdings noch als Zwischenschritt zur Ermittlung des PL.

Diese Norm stellt an die Kategorie 2 höhere Anforderungen als die DIN EN 954-1. Sie deckt sich in diesen Anforderungen mit dem BGIA-Report (siehe Kapitel 12.1). In der prEN ISO 13849-1 wird ein konkretes Schaltungsbild für Kategorie 2 Schaltungen gezeigt, das hier übersetzt und hinsichtlich der genannten Beispiele an den konkreten Fall dieser Diplomarbeit angepasst wurde:

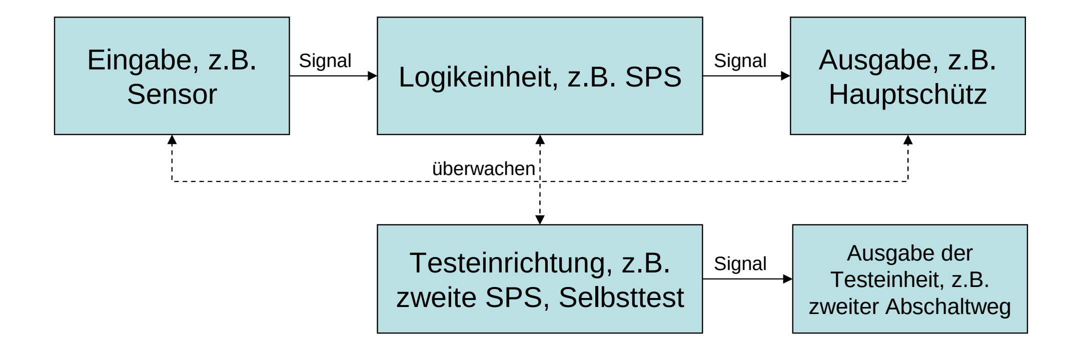


Abbildung 71: Kategorie 2 Schaltung umgesetzt aus Kapitel 6.2.5 der prEN ISO 13849 [15]

Diese Abbildung stellt, auch wenn sie grafisch von der Abbildung 70 (BGIA-Report) abweicht, die gleichen Funktionsanforderungen dar. Um Kategorie 2 nach prEN ISO 13849-1 zu erreichen müssen alle Tests aus Kapitel 10, wie im vorherigen Kapitel 12.1 beschrieben, umgesetzt werden.

Ohne die Verwendung einer Testeinrichtung wird nach prEN ISO 13849-1 nur die Kategorie B (niedrigste Stufe) erreicht, da die Kategorie 1 den Einsatz bewährter Bauteile verlangt, zu denen die Standard-SPS nicht zählt. Ein Einsatz der Standard-SPS mit Diagnosefunktionen in Sicherheitssteuerungen ohne eine Testeinrichtung bringt nach dieser Norm keine Verbesserung gegenüber der normalen Standard-SPS und wird aus diesem Grund hier nicht weiter betrachtet.

Die Einstufung der Steuerung in die Kategorien ist in der prEN ISO 13849-1 allerdings nur ein Zwischenschritt. Die letztendliche sicherheitstechnische Einstufung richtet sich, wie in Kapitel 5.2.1 beschrieben, nach dem erreichten Performance Level.

Um diese Einstufung an einem konkreten Beispiel darzustellen, werden in dieser Arbeit für eine Beispiel-SPS bauteiltypische Werte bezüglich deren Ausfallrate angenommen. Zu diesem Ansatz siehe auch die unveröffentlichte Präsentation des BGIA [5].

Die Werte der Beispiel-SPS sind der Siemens Norm 29500 Teil 2 [18] entnommen, die sich mit der Ausfallrate von Bauelementen beschäftigt. Die Ausfallraten in dieser Norm sind für eine mittlere Umgebungstemperatur von 40 °C und nach DIN EN 60721-3 definierten Umwelt-, Transport- und Lagerbedingungen angegeben.

Ausfallraten in dieser Siemens Norm sind in „FIT“ angegeben, wobei ein FIT dem Ausfall eines Bauteils pro 10^9 Bauelementstunden entspricht. Die Umrechnung in $MTTF_d$ bei Dauerbelastung lautet daher:

$$MTTF_d = \frac{10^9\,Stunden}{FIT * 365 Tage * 24 \frac{Stunden}{Tag}}$$

Abbildung 72: Umrechnung FIT in $MTTF_d$

Die Werte in der folgenden Tabelle wurden aus der o. a. Siemens Norm entnommen und mit 5 multipliziert. Dieser Multiplikator ergibt sich aus den Faktoren „10“ für den Worst Case und „½“ für

die Umrechnung von MTTF nach $MTTF_d$ aus der prEN ISO 13849-1. (siehe hierzu auch Kapitel 5.2.2)

Sicherheitsrelevante Blöcke der Beispiel-SPS	**FIT [1 pro 10^9h]**	**$MTTF_d$ [y]**
Ein dynamischer RAM Baustein mit 2 bis 4 Mbit	75	1522
Ein Flashspeicher mit 8 bis 16 Mbit	500	228
Eine 16 bit CPU mit 50k bis 500k Transistoren	1000	114
Platine, Verbindungstechnik, Lötstellen	100	1142
Watchdog, Unterspannungsüberwachung	50	2283
Zwei Eingänge, Optokoppler	400	285
Ein Ausgang, Optokoppler, 500 mA Treiber	200	571

Nach der Formel in Abbildung 73 ergibt sich eine gesamt $MTTF_d$ für eine solche Standard-SPS von 49 Jahren, was nach der prEN ISO 13849-1 (siehe auch Abbildung 17) die Einstufung „hoch“ bedeutet.

$$\frac{1}{MTTF_{d,gesamt}} = \sum_{i=1}^{N} \frac{1}{MTTF_{d,i}}$$

Abbildung 73: Formel zur Berechnung des gesamt MTTFd in Reihe geschalteter Bauteile aus prEN ISO 13849-1 Anhang D.1 [15] (siehe auch Abbildung 14)

Um den durchschnittlichen DC zu bestimmen müssen zunächst die DCs der einzelnen Blöcke ermittelt werden. Diese ergeben sich, wie folgt, aus den den Blöcken zugeordneten Tests:

- Der RAM Baustein wird in diesem Fall zur Hälfte mit dem Test des Bereichs der globalen Variablen (Kapitel 10.6.2 – DC 90%), und zur anderen Hälfte mit dem Test der benutzten Variablen (Kapitel 10.6.1 – DC 60%) getestet. Daraus ergibt sich für diesen Baustein ein durchschnittlichen DC von 75%.
- Für den Flashspeicher wird eine CRC Prüfung (Kapitel 10.5.1) mit 90% Diagnosedeckungsgrad angenommen.
- Der Prozessor wird durch die Ablaufkontrolle (Kapitel 10.3 – DC 90%) und den Prozessorselbsttest (Kapitel 10.4 – DC 60%) überwacht. Weiterhin testet auch die Funktionskontrolle durch die zweite SPS (Kapitel 0 – DC 99%) das Funktionieren des Prozessors. Deshalb kann ein kombinierter DC von 90% angenommen werden.
- Ein Ausfall der Platine würde zu einem Ausfall eines angeschlossenen Blocks führen. Da alle angeschlossenen Blöcke überwacht werden, wird angenommen, dass die Platine im Schnitt den gleichen DC wie die restlichen Teile besitzt. Aus diesem Grund kann sie für die Berechnung des durchschnittlichen DC vernachlässigt werden, da hierbei nur ein gewichteter Mittelwert der DCs berechnet wird.
- Der Watchdog selbst kann nicht überwacht werden. Er hat daher einen DC von 0%.
- Die Eingänge werden in diesem Beispiel durch den Test der redundanten Eingänge (Kapitel 10.7.2) getestet. Da sie regelmäßig angesprochen werden, erhalten sie danach einen DC von 90%.

- Die Ausgänge werden durch den Test über rückgekoppelte Eingänge (Kapitel 10.7.3) ständig überwacht und haben somit einen Diagnosedeckungsgrad von 99%.

Aus der nachfolgenden Formel

$$DC_{avg} = \frac{\sum_{i=1}^{N} \frac{DC_i}{MTTF_{d,i}}}{\sum_{i=1}^{N} \frac{1}{MTTF_{d,i}}} = MTTF_{d,gesamt} * \sum_{i=1}^{N} \frac{DC_i}{MTTF_{d,i}}$$

Abbildung 74: Formel zur Berechnung des durchschnittlichen DCs, nach prEN ISO 13849-1 Anhang E.2 [15], $MTTF_{d,\,gesamt}$ aus Abbildung 14 (siehe Abbildung 19)

ergibt sich ein durchschnittlicher Diagnosedeckungsgrad

$$DC_{avg} = 49 Jahre * (\frac{75\%}{1522 Jahre} + \frac{90\%}{228 Jahre} + \frac{90\%}{114 Jahre} + \frac{0\%}{2283 Jahre} + \frac{90\%}{285 Jahre} + \frac{99\%}{571 Jahre} = 88\%$$

von 88%. Nach der Tabelle in Abbildung 20 liegt dieser an der oberen Grenze von „niedrig“.

Angewendet auf die Tabelle in Abbildung 75 wird mit einer Kategorie 2 Steuerung, einem niedrigen DC und einer hohen $MTTF_d$ ein Performance Level von „c“ erreicht. In der Praxis könnte durch andere Bauteile mit geringeren Ausfallraten und / oder besseren Tests (z. B. CoDeSys-spezifischer Test aus Kapitel 10.6.3) auch ein DC von „mittel“ erreicht werden. Andererseits könnte beim Einsatz von Bauteilen minderer Qualität die $MTTF_d$ in dem Bereich „mittel“ liegen. Somit ergibt sich wie in Abbildung 75 dargestellt für die in diesem Kapitel behandelten SPSen mit Diagnosefunktion und zusätzlicher Testeinrichtung ein Spielraum für das PL von „b“ bis „d“.

Kategorie	B	1	2	2	3	3	4
DC_{avg}	keine	keine	niedrig	mittel	niedrig	mittel	hoch
$MTTF_d$ jedes Kanals: niedrig	a	Nicht abgedeckt	a	b	b	c	Nicht abgedeckt
$MTTF_d$ jedes Kanals: mittel	b	Nicht abgedeckt	b	c	c	d	Nicht abgedeckt
$MTTF_d$ jedes Kanals: hoch	Nicht abgedeckt	c	c	d	d	d	e

Standard-SPS

Sicherheits-SPS

Standard-SPS mit Diagnosefunktion

Mögliche Schwankungen

Abbildung 75: prEN ISO 13849-1 Tabelle 7: Vereinfachte Bestimmung des Performance Levels [15] (siehe Abbildung 21) mit Eingetragener Auswahl für Standard-SPS, Sicherheits-SPS und Standard-SPS mit Diagnosefunktion (Tests nach Kapitel 10)

Nach prEN ISO 13849-1 muss beim Einsatz einer Kategorie 2 Steuerung eine Beurteilung der Fehler gemeinsamer Ursachen (CCF) vorgenommen werden und es muss sichergestellt sein, dass die Addition dieser mit Punkten bewerteten Maßnahmen 65 Punkte ergibt. Siehe hierzu Kapitel 5.2.3, Abbildung 22. Diese Bewertung kann nur im konkreten Einzelfall durchgeführt werden.

Weiterhin müssen zum Erreichen der Kategorie 2 nach prEN ISO 13849-1 folgende Bedingungen erfüllt sein:

- Die Testrate muss mindestens 100 mal größer sein als die Anforderungsrate der Sicherheitfunktion.
- Die $MTTF_d$ der Testeinrichtung (hier SPS2) muss mindestens halb so lang sein, wie die der Logikeinheit (SPS1).

13 Zusammenfassung

Im Steuerungsbau werden regelmäßig Speicherprogrammierbare Steuerungen verwendet. Diese SPSen können wegen ihrer niedrigen sicherheitstechnischen Einstufung nach den gesetzlichen Anforderungen an Sicherheitssteuerungen im europäischen Binnenmarkt nur stark eingeschränkt und dann auch nur in Verbindung mit zusätzlichen Maßnahmen verwendet werden. In dieser Arbeit wird aufgezeigt, wie die Steuerungskategorie einer Standard-SPS nach DIN EN 954-1, bzw. ihr Performance Level nach prEN ISO 13849-1 mit Hilfe von zusätzlichen Programmbausteinen erhöht werden kann, so dass sie in großen Anwendungsbereichen für Sicherheitssteuerungen verwendet werden kann.

Weiterhin wird die Einordnung der Sicherheitssteuerungen in die rechtlichen Vorgaben des europäischen Binnenmarktes – und hier, wegen ihrer zentralen Bedeutung, insbesondere in die Vorgaben der Maschinenrichtlinie – dargestellt.

13.1 Ergebnis

Die in Kapitel 7.1.1 aufgezeigte Kategorie B der Standard-SPS nach DIN EN 954-1 kann wie in Kapitel 12.1 dargelegt, mit Hilfe der in der Diplomarbeit erarbeiteten Funktionsbausteinen, auf Kategorie 2 dieser Norm erhöht werden.

Der in Kapitel 7.1.2 aufgezeigte Performance Level „a“ nach prEN ISO 13849-1 kann wie in Kapitel 12.2 erläutert, abhängig vom konkreten Einzelfall, in den Bereich der Performance Level „b“ bis „d“ der vorgenannten Norm angehoben werden.

13.2 Praktischer Nutzen

Der praktische Einsatz einer Standard-SPS mit Diagnosefunktionen ist im Rahmen einer Maschinensteuerung hauptsächlich in den drei nachfolgenden Fällen möglich:

- Einsatz der SPS im Neumaschinenbau
- Sicherheitstechnische Aufrüstung einer vorhandenen Standard-SPS einer im Einsatz befindlichen Maschine / Gebrauchtmaschine auf den heutigen Stand der Sicherheitstechnik
- Einsatz einer Standard-SPS mit Diagnosefunktion im Rahmen eines Steuerungs-Umbaus bei im Einsatz befindlichen Maschinen / Gebrauchtmaschinen

Nach Expertenschätzung (BGIA) liegen zur Zeit 50% aller neuen Maschinen in dem sicherheitstechnischen Bereich der durch eine nach dieser Diplomarbeit erweiterte Standard-SPS abgedeckt werden kann. Diese Standard-SPS kann hier eine wesentlich wirtschaftlichere Lösung gegenüber dem Einsatz einer teuren und sicherheitstechnisch überqualifizierten Sicherheits-SPS bieten.

In der Industrie wurden in der Vergangenheit zum Teil Standard-SPSen ohne besondere Diagnosefunktionen in sicherheitstechnischen Steuerungen eingesetzt. Zum Beispiel sind in auf dem

Markt SPSen bekannt, die – regelwidrig – die Inertisierung zur Verhinderung einer explosionsgefährdeten Atmosphäre steuern. Die nachträgliche sicherheitstechnische Qualifizierung einer solchen SPS mit den in dieser Diplomarbeit beschriebenen Funktionsbausteinen kann ggf. die notwendige Sicherheit herstellen. In diesem Fall ist die notwendige Erweiterung der Programmierung der SPS wesentlich kostengünstiger als der komplette Austausch der Steuerung durch eine Sicherheits-SPS. Dazu kommt, dass der Eingriff an den beim Kunden befindlichen Maschinen und Anlagen schneller vorgenommen werden kann, da lediglich die externe Verdrahtung der SPS minimal ergänzt werden muss.

Ältere im Betrieb befindliche Maschinen verfügen häufig noch über fest verdrahtete Relaissteuerungen. Solche Steuerungen werden zunehmend durch modernen SPSen ersetzt. Auch hierbei kann der Einsatz von Standard-SPSen, die mit den in dieser Diplomarbeit erarbeiteten Funktionsbausteine erweitert worden sind, häufig die wirtschaftlichere Lösung gegenüber dem Einsatz einer Sicherheits-SPS sein.

13.3 Ausblick

13.3.1 Fehlernummern speichern

Eine sinnvolle Erweiterung der Testbausteine, die aus Zeitgründen wegfallen musste, ist ein Speichern von Fehlernummern vor einem Abschalten. Dadurch ist es dem Benutzer der SPS schnell möglich die Ursache für den Ausfall der SPS festzustellen.

13.3.2 Erhöhen der niedrigen Diagnose Deckungsgrade (DCs)

Einige der in dieser Diplomarbeit erarbeiteten Tests haben auf Grund der stark eingeschränkten Programmier-Möglichkeiten durch die Vorgaben der Norm DIN EN 61131 zur SPS-Programmierung [16] einen relativ „niedrigen“ DC von 60%. Das bedeutet, dass 40% der Fehler bei diesen Tests unerkannt bleiben. Eine Erhöhung dieser niedrigen DCs würde zu einer Verbesserung des Gesamt-DC der Steuerung führen.

In Kapitel 10.6.3 wird demonstriert, wie z. B. mit einem gegenüber der o. a. Norm erweiterten Befehlsumfang der in der Norm vorgegebenen Programmiersprache der DC für den Speichertest von 60% auf 90% erhöht werden kann.

13.3.3 Verbesserung der Wirksamkeit der Speichertests durch Hardwareinformationen

Wie in Kapitel 9.1 beschrieben kann es beim Abspeichern von Daten zu Übertragungsfehlern kommen, indem das Signal zusätzlich in benachbarten Speicherzellen abgespeichert wird. Das Aufdecken von Übertragungsfehlern wird durch die Speicher-Tests in den Kapiteln 10.6.2 und 10.6.3 erreicht, indem ein möglichst breiter Bereich des Speichers mittels Signatur getestet wird. (siehe Abbildung 76) Dabei gehen die Tests davon aus, dass das physikalische Layout des Speichers nicht bekannt ist. Das physikalische Layout meint dabei die Anzahl der Speicherzellen pro Reihe und die

Gesamtzahl der Reihen. Aus diesem Grund muss eine relativ große Menge an Speicherplätzen überwacht werden.

Hierbei erhöht sich mit zunehmender Größe des getesteten Bereichs die Wahrscheinlichkeit, dass der Bereich des Speichers, in dem ein Übertragungsfehler geschehen kann, sich unter den überwachten Bereichen befindet. D. h. für eine hohe Sicherheit muss ein großer Bereich überwacht werden.

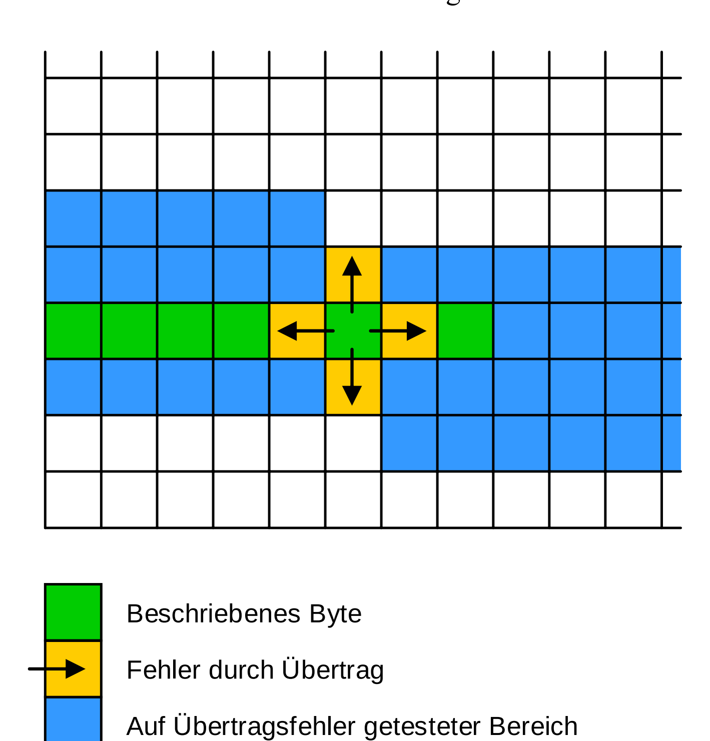


Abbildung 76: Speichertest mit breitem Testbereich

Ist jedoch das physikalische Layout des Speichers bekannt, so kann der Test auf ein Minimum an Aufwand reduziert werden, bzw. die Güte des Tests bei gleichem Aufwand erhöht werden, da der getestete Bereich optimal um den beschriebenen Bereich platziert werden kann. (siehe Abbildung 77)

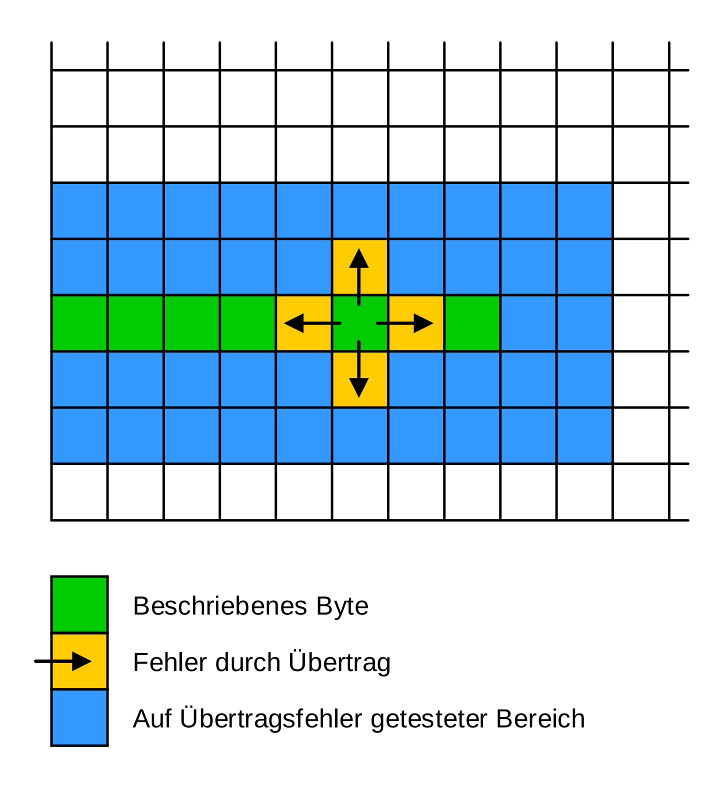


Abbildung 77: Speichertest mit passendem Testfeld

13.3.4 Möglichkeiten für weitere Tests der Ein- und Ausgabehardware

13.3.4.1 Doppelter Eingang

Hierbei werden zwei Eingänge durch zusätzliche Verkabelung parallel geschaltet und durch einen Funktionsbaustein auf Gleichheit der gelesenen Eingangssignale überwacht. Wie dieser Test ggf. durchgeführt werden könnte, ist in Kapitel 10.7.2 beschrieben.

13.3.4.2 Dynamisches Signal zum Test von Ein- und Ausgängen

Teile der Hardware der Eingänge von SPSen (siehe Kapitel 6.2.3) werden von allen Eingängen einer Eingangskarte gemeinsam genutzt. Ein Ausfall eines solchen mehrfach genutzten Hardwarebauteils wirkt sich daher auf mehrere Eingänge aus. Ein Ausfall eines solchen Bauteils führt deshalb zum Ausfall der gesamten Eingangskarte. Gleiches gilt auch für solche Bauteile in der Ausgangskarte der SPS.

Bei diesem Test wird ein Testsignal von einem Ausgang zurück zu einem Eingang der SPS geschickt und durch den Prozessor mit dem erwarteten Ergebnis verglichen. Der dafür genutzte Aus- bzw. Eingang der SPS, wird durch diesen Test belegt und kann somit keine andere Funktion wahrnehmen. Wird ein fehlerhaftes Signal festgestellt, wird aus Sicherheitsgründen von einem Defekt auch an den benachbarten Ein- und Ausgängen ausgegangen.

Diese Methode hat den Nachteil, dass sie nur sinnvoll benutzt werden kann, wenn in den Ein- und Ausgangskarten möglichst viele Bauteile gemeinsam genutzt werden, da nur diese überwacht werden. Der DC des Tests hängt deshalb von der prozentualen Anzahl dieser Bauteile an der Anzahl der Gesamtbauteile ab, da nur gemeinsam genutzte Hardware überwacht wird. Fehlalarme sind ebenfalls

möglich, wenn Hardware des Testeingangs ausfällt, die von den anderen Ein- und Ausgängen nicht genutzt wird uns somit zu keinem gefährlichen Fehler in anderen Teilen der SPS führt.

Der Vorteil hingegen liegt in dem sehr geringen Aufwand an zusätzlicher externer Beschaltung und, durch das gleichzeitige Überwachen aller Ein- bzw. Ausgänge einer Karte, in einer relativ hohen Testgeschwindigkeit. Beim Einsatz mehrerer Ein- und Ausgangskarten an der SPS muss jede Karte einzeln geprüft werden.

13.3.5 Erstellung der Testverfahren durch den Hersteller des Compilers

Die in dieser Diplomarbeit beschriebenen Tests können wesentlich an Effektivität gewinnen, wenn der vom Anwender genutzte Compiler (Übersetzer der Programmiersprache in den Maschinencode der SPS) diese Testverfahren vorprogrammiert enthält und sie direkt in den auszuführenden Code implementiert. In Zusammenhang mit der Programmierung des Compilers können die Testverfahren erheblich näher an der Hardware programmiert werden. Dazu kommt, dass der eingeschränkte Befehlsumfang der DIN EN 61131-3 dann keine Rolle mehr spielen würde.

Komplizierte Vorgänge, wie das Implementieren der Ablaufkontrolle in das lauffähige Programm, können vom Compiler meist besser erledigt werden, als vom Anwender. Die Implementierung der Testroutinen in die Anwenderprogramme der SPS wird dadurch weniger Fehleranfällig, da der Anwender bei der Programmierung der SPS nur noch die Parameter der Testverfahren auszuwählen hat.

Allerdings ist beim Anwenden dieses Verfahrens zu beachten, dass der Hersteller des Compilers die Verantwortung für die richtige Funktion der von ihm auf den Markt gebrachten Testverfahren übernimmt.

13.3.6 Synchronisierung zweier SPSen zum zweikanaligen Abarbeiten von Sicherheitsprogrammen

Um die höchsten Steuerungskategorien bzw. Performance Level im Steuerungsbau zu erreichen ist es notwendig die gesamte Steuerung zweikanalig aufzubauen. Dies gilt auch für eine in der Steuerung enthaltene SPS. Das heißt es müssen in diesem Fall zwei SPSen parallel geschaltet werden. Beide SPSen arbeiten dann nebeneinander autark und vergleichen ständig an verschiedenen Stellen ihre aktuellen Daten.

Dabei ist das Problem des Datenaustauschs zu berücksichtigen. Dieses entsteht größtenteils durch das EVA Prinzip der SPS, wodurch ein Austausch von Daten selbst bei synchron arbeitenden SPSen zwei Zyklen dauert. Das bedeutet, es dauert zwei Zyklen zum Vergleich der gelesenen Eingänge und zwei Zyklen zum Vergleich der zu schreibenden Ausgänge um einen normalen Zyklus einer alleine arbeitenden SPS abzuarbeiten. Bei nicht synchronisierten SPSen dauert dieser Vorgang entsprechend länger.

Zum zweikanaligen Arbeiten wird aus diesem Grunde entweder ein Eingriff in die vorhandene Hardware der SPSen angewendet, der eine externen Synchronisation der SPSen verbunden mit einem

externen Datenvergleich durchführt (siehe hierzu Kapitel 6.1.4), oder es werden hohe Performanceverluste durch den notwendigen Datenaustausch in Kauf genommen.

Eine weitere Möglichkeit ist das nachträgliche Vergleichen von gesetzten Ergebnissen. Die SPSen merken sich z. B. die letzten 20 Ausgangszustände der letzten 20 Zyklen und führen untereinander einen blockweisen Vergleich eines Teiles dieser Ausgangszustände durch. Hierbei muss beachtet werden, dass ein gefundener Fehler bereits 20 Zyklen zurückliegen kann und zu diesem Zeitpunkt bereits eine kritische Situation herbeigeführt hat. (siehe hierzu Echtzeitfähigkeit in Kapitel 6.5)

13.3.7 Übertragung der erarbeiteten Testverfahren in andere Programmiersprachen

Die Quelltexte der erarbeiteten Testverfahren sind in der Normsprache AWL der Norm DIN EN 61131-3 [16] geschrieben. Nicht alle auf dem Markt erhältlichen SPSen sind in dieser Sprache programmierbar. Hier sind als Beispiel die weit verbreiteten SPSen von Siemens zu nennen. Diese können jedoch auch in einer AWL ähnlichen Sprachen programmiert werden. Es ist deshalb möglich, die Testverfahren für Siemens-SPSen umzuschreiben.

Grundsätzlich muss darauf geachtet werden, dass die in dieser Diplomarbeit erarbeiteten Testverfahren an der AWL Sprache der o. a. Norm optimiert wurden. Einige in dieser Diplomarbeit verwendeten Programmiertaktiken, z. B. das Verwenden des aktuellen Ergebnisses über Sprungbefehle hinaus, können in anderen Sprachen nicht verwendet werden. Ein Übersetzen der programmierten Tests in eine Sprache, die keine Sprungmarken zulässt wird sich deshalb als schwierig erweisen. In solchen Fällen ist es anzuraten die Tests auf Basis der beschriebenen Verfahren neu zu programmieren und nicht auf Basis deren Quelltexte umzuschreiben.

14 Abbildungsverzeichnis

15 Abkürzungsverzeichnis

3S	Firma: Smart Software Solutions
73/23/EWG	Richtlinie Nummer 23 aus dem Jahre 1973, Europäische Wirtschaftsgemeinschaft
98/37/EG	Richtlinie Nummer 37 aus dem Jahre 1998, Europäische Gemeinschaft
A	Ampere
AE	Aktuelles Ergebnis
ALU	Aritmethical Logical Unit
AS	Ablaufsprache
ATEX	Atmosphärischer Explosionsschutz
AWL	Programmiersprache: Anweisungsliste
B_{10d}*-Wert*	Schaltspieleanzahl, bei der statistisch 10% der Stichproben ausfallen
BGIA	Berufsgenossenschaftliches Institut für Arbeitsschutz
BIA	Berufsgenossenschaftliches Institut für Arbeitssicherheit
Bit	Kleinste Einheit eines binären Speichers. Kann nur Eins oder Null enthalten.
Byte	Speicherstelle aus Acht Bit
CAA	CoDeSys Automation Alliance
CCF	Ausfall in Folge gemeinsamer Uhrsachen (Common Cause Failure)
CE	Europäische Übereinstimmung (franz.: Conformité Européen)
CEN	Europäisches Komitee für Normung (franz.: Comité Européen de Normalisation)
CISC	Complex Instruction Set Computing
CoDeSys	Code Development System
CRC	Zyklische Redundanz Prüfung (Cyclical Redundancy Check(ing))
DC	Fehleraufdeckungsrate (Diagnostic Coverage)
DC_{avg}	Fehleraufdeckungsrate im Durchschnitt (average Diagnostic Coverage)
DIN	Deutsche Industrie Norm
EEPROM	Elektrisch löschbarer programmierbarer nur lesbarer Speicher (Electrically erasable programmable read only memory)
EFTA	Europäische Freihandelszone
EG	Europäische Gemeinschaft
EMV	Elektromagnetische Verträglichkeit
EMVG	Gesetz über die Elektromagnetische Verträglichkeit
EMV-RL	Richtlinie über die Elektromagnetische Verträglichkeit

EN	Europäische Norm
EU	Europäische Union
EVA	Eingabe Verarbeitung Ausgabe
EWG	Europäische Wirtschaftsgemeinschaft
EWR	Europäischer Wirtschaftsraum
F1	Seltene bis öfter und/oder kurze Dauer der Gefährdungsexposition
F2	Häufig bis dauernd und/oder lange Dauer der Gefährdungsexposition
FBS	Funktionsbausteinsprache
FUB	Funktionsbaustein
GPSG	Geräte- und Produktssicherheitsgesetz
h	Stunde (hour)
HVBG	Hauptverband der gewerblichen Berufsgenossenschaften
KOP	Kontaktplan
MRA(´s)	Abkommen über die gegenseitige Anerkennung von benannten Prüfstellen (Mutual Recognition Agreement)
MRL	Maschinenrichtlinie
MTBF	Mittlere Ausfallzeit, die im normalen Betrieb vergeht, bevor ein Fehler auftritt (Mean Time Between Failures)
MTTF	Mittlere Zeit bis zum Ausfall (Mean Time To Failure)
$MTTF_d$	Mittlere Zeit bis zum gefährlichen Ausfall (Mean Time To dangerous Failure)
P1	Vermeidung der Gefährdung unter bestimmten Bedingungen möglich
P2	Vermeidung der Gefährdung kaum möglich
PC	Privater Rechenautomat (Personal Computer)
PFH	Wahrscheinlichkeit eines Ausfalls pro Stunde (Probability of Failure per Hour)
PFH_d	Wahrscheinlichkeit eines gefährlichen Ausfalls pro Stunde (Probability of dangerous Failure per Hour)
PL	Performance Level
PL_r	Benötigter Performance Level (Performance Level required)
POE	Programmorganisationseinheit
prEN	Vorläufige Version einer Europäischen Norm (pre European Norm)
QS	Qualitätssicherung
RAM	(Random Access Memory)
RC-Filter	Filter bestehend aus Wiederständen und Kondensatoren

RISC	Reduced Instruction Set Computing
S1	Leichte (üblicherweise reversible) Verletzung
S2	Schwere (üblicherweise irreversible) Verletzung, einschließlich Tod
SIL	Sicherheits-Integritätslevel (Savety Integrity Level)
SPS	Speicherprogrammierbare Steuerung
SRP/CS	Sicherheitsbezogener Teil einer Steuerung (Safety Related Parts of a Control System)
ST	Strukturierter Text
V	Volt

16 Literaturverzeichnis

16.1 Bücher und Zeitschriften

[1] Berufsgenossenschaftliches Institut für Arbeitsschutz (2006): Selbsttests für Mikroprozessoren mit Sicherheitsaufgaben oder: „Quo vadis Fehler?“, BGIA Report #/2006, St. Augustin (noch nicht veröffentlicht)

[2] Bömer, Dipl.-Ing. Thomas, Büllesbach, Dipl-Ing. Karl-Heinz (2006): Neuer Ansatz für die Sicherheit von Maschinen: prEN ISO 13849-1 – Sicherheitsbezogene Teile von Steuerungen, Beilage aus MRL-News – 22/03/06 Wuppertal (SCHMERSAL Holding GmbH & Co. KG)

[3] Europäische Kommission (2000): Leitfaden für die Umsetzung der nach dem neuen Konzept und dem Gesamtkonzept verfassten Richtlinien (auch Binnenmarktleitfaden oder Blue Guide genannt), Luxemburg (Amt für amtliche Veröffentlichungen der Europäischen Gemeinschaften)

[4] Gall, H., Kemp, K. (1997): Wirksamkeit von zeitlichen und logischen Programmlaufüberwachungen beim Betrieb von Rechnersystemen, Dortmund / Berlin (Schriftenreihe der Bundesanstalt für Arbeitsschutz und Arbeitsmedizin)

[5] Huelke, Dr. Michael (2006): „Zuverlässigkeit von Standard-Steuerungen“, Fachgespräch Maschinen- und Gerätesicherheit am 08./09.03.2006, Präsentation beim BGIA

[6] Kleinbreuer, Werner, Kreutzkampf, Franz, Meffert, Karlheinz, Reinert, Dietmar (1997): Kategorien für Steuerungen, BIA Report 6/97, St. Augustin (HVBG)

[7] Krämer, Manfred (1990): Speicherprogrammierbare Steuerungen in der Sicherheitstechnik, Der Elektroniker 10/1990, Stuttgart

[8] John, K.-H., Tiegelkamp, M. (1999): SPS-Programmierung mit IEC 61131-3, Berlin Heidelberg (Springer)

[9] Ostermann, Dipl.-Ing. Hans-J. (2002): Einführung in den Binnenmarkt, aus Sicher ist Sicher, Heft 3/2002, Berlin (K.L.U.G.E. Verlag)

[10] Ostermann, Dipl.-Ing. Hans-J. (1995): EG-Maschinenrichtlinie …Was ist zu tun? aus Die BG, Heft 7/1995, Berlin (Erich Schmidt Verlag GmbH & Co)

[11] Ostermann, Dipl.-Ing. Hans-J., von Locquenghien, Dipl.-Ing. Dirk (2006): Wegweiser Maschinensicherheit, Stand 29. Ergänzungslieferung, Februar 2006 (Bundesanzeiger Verlag)

[12] Pöppinghaus, Wolfgang (1999): dtv-Atlas Weltgeschichte, München (Deutscher Taschenbuch Verlag GmbH & Co. KG)

[13] Seitz, Mathias (2003): Speicherprogrammierbare Steuerungen, München (Fachbuchverlag Leipzig)

16.2 Normen

[14] DIN EN 954-1 Sicherheitsbezogene Teile von Steuerungen, 1996(D)

[15] prEN ISO 13849-1 Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design, 2005(E)

[16] DIN EN 61131 Teil 1 bis 7 Speicherprogrammierbare Steuerungen, 2003(D)

[17] DIN EN 61508 Teil 1 bis 7 Funktionale Sicherheit sicherheitsbezogener elektrischer/ elektronischer/ programmierbarer elektronischer Systeme, 2001(D)

[18] SN 29500 Teil 2 Ausfallraten Bauelemente, 1999(D), München, Siemens AG

16.3 Richtlinien und Gesetze

[19] EWG-Vertrag von 1997 (in Kraft getreten 1958) geändert durch die Einheitliche Europäische Akte (in Kraft getreten in 1987), durch den Maastricher Vertrag über die Europäische Union (in Kraft getreten in (1993) und durch den Amsterdamer Vertrag (in Kraft getreten 1999)

[20] Richtlinie des Rates zur Angleichung der Rechtsvorschriften der Mitgliedsstaaten betreffend elektrische Betriebsmittel zur Verwendung innerhalb bestimmter Spannungsgrenzen (73/23/EWG - Niederspannungsrichtlinie)

[21] Richtlinie des Rates zur Angleichung der Rechtsvorschriften der Mitgliedsstaaten für einfache Druckbehälter (87/404/EWG – Richtlinie einfache Druckbehälter)

[22] Richtlinie des Rates zur Angleichung der Rechtsvorschriften der Mitgliedsstaaten über die elektromagnetische Verträglichkeit (89/336/EWG - EMV-Richtlinie)

[23] Beschluss 93/465/EWG über die in den technischen Harmonisierungsrichtlinien zu verwendenden Module für die verschiedenen Phasen der Konformitätsbewertungsverfahren und die Regeln für die Anbringung und Verwendung der CE-Konformitätskennzeichnung

[24] Richtlinie 94/9/EG des Europäischen Parlaments und des Rates zur Angleichung der Rechtsvorschriften der Mitgliedstaaten für Geräte und Schutzsysteme zur bestimmungsgemäßen Verwendung in explosionsgefährdeten Bereichen

[25] Richtlinie 97/23/EG des Europäischen Parlaments und des Rates zur Angleichung der Rechtsvorschriften der Mitgliedstaaten über Druckgeräte (Druckgeräterichtlinie)

[26] Richtlinie 98/37/EG des Europäischen Parlaments und des Rates zur Angleichung der Rechts- und Verwaltungsvorschriften der Mitgliedstaaten für Maschinen (Maschinenrichtlinie)

[27] Richtlinie 2006/42/EG des Europäischen Parlaments und des Rates über Maschinen und zur Änderung der Richtlinie 95/16/EG (Neufassung) (neue Maschinenrichtlinie)

[28] Gesetz über technische Arbeitsmittel und Verbraucherprodukte (Geräte- und Produktsicherheitsgesetz – GPSG), BGBl. I vom 9. Januar 2004

[29] Verordnung zum Schutz vor Gefahrstoffen (Gefahrstoffverordnung - GefStoffV) vom 23. Dezember 2004 (BGBl. I S 3758), geändert durch Artikel 2 der Verordnung vom 23. Dezember 2004 (BGBl. I S 3855)

16.4 Webseiten

[30] http://www.maschinenrichtlinie.de, herunter geladen am 1. Mai 2006

[31] http://www.wikipedia.de, herunter geladen am 1. Mai 2006

[32] http://www.speed7.com/, herunter geladen am 22. Juni 2006